



产品手册
机器人系统操作说明书



产品手册

机器人系统操作说明书

V2.3

适用于 IRC 系列控制柜

软件版本：V7.6

示教器系统操作支持六种语言：中/英/韩/俄/越/日

本手册旨在为您提供产品相关的支持性信息，其内容可能随时更新，恕不另行通知。

捷勃特力求手册内容的准确性与时效性，但其中所载信息不构成捷勃特的任何明示担保或承诺。

请您知悉，对于因使用、参考或依赖本手册内容而可能引起的任何损害（包括但不限于直接、间接、偶然或结果性损害），捷勃特概不承担任何责任。

在任何情况下，未经捷勃特事先书面许可，不得复制或传播本手册的任何部分。

如需获取本手册的更多副本，请与捷勃特联系。

如在使用本手册过程中给您带来不便，敬请谅解！

注：本出版物以中文版本为准。

本出版物全部为国际单位制，GB 代表中国国家标准。

©版权所有 2026 捷勃特保留所有权利。

上海捷勃特机器人有限公司

中国上海

修订

版本号	修订时间	状态
V1.0	2023.02.25	取消
V1.1	2023.04.03	取消
V1.2	2023.07.13	取消
V2.0	2023.07.28	取消
V2.2	2025.07.24	草稿
V2.3	2026.05.22	发布

安全使用须知

在使用机器人时，必须熟读并理解本章所述内容。

在本操作手册中，机器人系统是指包含工业机器人本体、工业机器人控制器、示教器、线缆、软件及其他配件的集成系统，必须充分考虑用户和系统的安全预防措施。

没有上海捷勃特机器人有限公司的授权，任何人无权对工业机器人进行改造。上海捷勃特机器人有限公司对使用任何非捷勃特机器人提供的额外组件（软件、工具等）可能导致的工业机器人或者部件损坏不承诺任何责任。

上海捷勃特机器人有限公司对任何错误使用工业机器人导致的结果不承诺任何责任。错误使用包括：

- 超出机器人指定参数范围使用
- 作为人或者动物的运载工具
- 作为攀爬工具使用
- 在爆炸性环境情况下使用
- 无安全防护情况下使用

除了本章的安全内容，本操作手册包含其他安全说明，这些也必须遵守。

使用者的定义

作业人员的定义如下所示：

- 操作员
进行机器人的电源 on/off 操作
从操作面板启动机器人程序
- 调试工程师
进行机器人的操作
在安全围栏内进行机器人的示教及程序编写调试
- 维修工程师
进行机器人的操作

在安全围栏内进行机器人的示教等

进行机器人的维护（修理、调整、更换）作业

“操作员”不能进入安全围栏内进行作业。

“调试工程师”、“维修工程师”可以在安全围栏内进行作业。

安全围栏内的作业，包括搬运、设置、示教、调整、维护等。

要在安全围栏内进行作业，必须接受过机器人的专业培训。

在进行机器人的操作、编程、维护时，操作者、程序员、维修工程师必须警告安全，至少应穿戴下列物品进行作业。

- 适合于作业内容的工作服
- 安全鞋
- 安全帽

作业人员系统权限

操作员

操作员权限包含：

1. 机器人的开关机操作
2. 使用手持示教器进行示教机器人；选择、调试运行、启动、暂停、终止程序
3. 通过示教器屏幕上方状态栏切换当前加载的 TF/UF、修改全局速度参数
4. 允许移动至目标点等操作
5. 查阅报警，复位常规报警
6. IO 状态界面和寄存器界面的操作



1. “操作人员”不能在安全栅栏内进行作业。
2. 安全栅栏内的作业，包括搬运、设置、示教、调整、维护等。
3. 如要在安全栅栏内进行作业，必须接受过机器人的专业培训。

4. 在进行机器人的操作、编程、维护时，操作者、编程人员、维护技术人员必须注意安全，必须穿戴防护用品，如适合于作业内容的工作服、安全鞋以及安全帽。

调试工程师

调试工程师权限包含：

1. 包含操作人员所有权限
2. 可进行机器人零点设置、软限位设置、坐标系的建立和编辑
3. I/O 的配置与管理
4. 通讯配置
5. 新建、编辑、修改、删除等机器人程序管理功能
6. 新建各类寄存器并设置
7. 机器人程序属性的管理功能
8. 程序启动方式设置
9. 文件的备份和加载
10. 设置控制柜主板 IP 地址/TP IP 地址
11. 设置系统时间

管理员

管理员权限包含：

1. 包含操作员、调试工程师的所有权限
2. 软件的安装升级
3. 对程序员角色的管理，可添加、删除、编辑程序员角色

有关安全的记载的定义

本说明书包括保证使用者人身安全以及防止机床损坏的有关安全的警告事项，并根据它们在安全方面的重要程度，在正文中以“危险”和“警告”来叙述。

此外，有关的补充说明以“注意”来叙述。

用户在使用之前，必须熟读“危险”、“警告”和“注意”中所叙述的事项。

标识	定义
----	----

 危险	用于在错误操作时，有可能会出现使用者死亡或者受重伤等危险的情况。
 警告	用于在错误操作时，有可能会出现人员轻度或中度受伤、物品受损等危险的情况。
 注意	用于记述补充说明，属于危险或者警告以外的事项。

请仔细阅读本说明书，为了方便随时参阅，请将其妥善保管在身边。

作业人员的安全

在自动运行机器人时，首先必须设法确保作业人员的安全。在机器人自动运行过程中，进入机器人的动作范围是十分危险的。应采取防止作业人员进入机器人动作范围的措施。

下面列出一般性的注意事项。请妥善采取确保作业人员安全的相应措施。

1. 运用机器人系统的各作业人员，应通过上海捷勃特机器人有限公司的课程。
2. 在设备运转之中，即使机器人看上去已经停止，也有可能是因为机器人在等待启动信号而处在即将启动的状态。即使在这样的情况下，也应该将机器人视为正在运动中。
3. 务必在机器人系统周围设置安全围栏和安全门。
4. 应尽可能将外围设备设置在机器人的动作范围之外。
5. 应根据需要设置锁具，使得负责操作的人员以外者，不能接通机器人的电源。
6. 在进行外围设备的个别调试时，务必断开机器人的电源后再执行。
7. 搬运或安装机器人时，务必按照上海捷勃特机器人有限公司所示的方式正确的进行。如果以错误的方法进行作业，则有可能由于机器人的翻倒而导致作业人员受伤。
8. 在安装好之后首次使机器人操作时，务必以低速进行。然后，逐渐的加快速度，并确认是否有异常。
9. 在使用机器人操作时，务必在确认安全围栏内没有人员后再进行操作。同时，检查是否存在潜在的危險，当确认存在潜在的危險时，务必排除危險之后再进行操作。
10. 不要在下面所示的情形下使用机器人。否则，不仅会给机器人造成不良影响，而且还可能导致作业人员受重伤。
 - a. 在有可燃性的环境下使用

- b. 在有爆炸性的环境下使用
- c. 在存在大量辐射的环境下使用
- d. 在水中或高湿度环境下使用
- e. 在连接与停止相关的外围设备和机器人的各类信号时，务必确认停止的作，以避免错误连接。

安全警告标签

机器人和控制器都贴有数个安全和信息标签，其中包含产品的相关重要信息。这些信息对所有操作机器人系统的人员都非常有用，如安装、检修或操作期间。

安全标签只使用图形，适用于所有语种。



必须遵守产品标签上的安全和健康标志。此外，还需遵守系统构建方或集成方提供的补充安全信息。

标志	描述
	如果在通电时触碰控制器内部通电零件，可能会触电。
	如果不依照说明操作，可能会发生能造成人员伤害或产品损坏的事故。适用于某些涉及功能要求的警告消息。
	控制柜接地标志

	<p>将手移至移动部件附近时可能会使手或手指卡在轴和外罩之间。</p> <p>装有伸缩罩的机器人不会夹住手或手指，因此没有该标签。</p>
	<p>请勿在机器人移动时进入作业区域内。机器人可能碰撞到作业人员。这非常危险，可能会造成严重的安全问题。</p>
	<p>高温，注意不要灼伤。</p>

机器人的停止

捷勃特机器人停止方式包含：

急停(相当于 IEC 60204-1 的类别 0 的停止)

- 断电停止即通过断开伺服电源，机器人瞬间停止的方式。
- 机器人立即抱闸，同时断开伺服电源。机器人动作将立即停止。

保护性停止(相当于 IEC 60204-1 的类别 1 的停止)

- 控制停止即机器人执行减速后，断开伺服电源的停止方式。
- 控制机器人减速，1s 后断开伺服电源。



控制停止的停止距离以及停止时间，要比断开电源停止更长。使用控制停止时，考虑到停止距离以及停止时间变长，需要对整个系统进行充分的风险评估和分析。

机器人急停及保护停止的策略

根据文件 GB 5226.1-2008 规定以下三类停止功能

0 类：用即刻切除机械制动机构动力的办法停止（急不可控停止）

1 类：给机械制动机构施加动力去完成停车并在停车后切除动力的可控停止

2 类：利用储留动能施加于机械制动机构的可控停止

捷勃特机器人的保护性停止属于 1 类停止的方式。

捷勃特机器人的急停属于 0 类停止的方式。

详情见下表

	急停	保护停止
场合	操作者有快速无障碍通道	由安全距离规则决定
启动	手动	自动或手动
安全系统性能	GB/T 16855.1-2008 中的类别 3, 或有风险评估决定	GB/T 16855.1-2008 中的类别 3, 或有风险评估决定
复位	只能手动	手动或自动
使用频率	不频繁；仅在紧急情况下使用	可变的；每个循环中使用或不频繁使用
作用	去除所有危险的能源	控制被防护的危险

安全功能

安全功能概览

捷勃特机器人配有以下安全功能：

- 紧急停止
- 使能装置
- 外部安全装置接口

- 外部紧急停止按钮接口
- 外部隔离性防护装置接口
- 外部限位停止装置接口

紧急停止

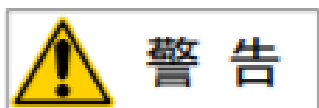
捷勃特机器人具有如下急停装置：

- 示教器右上的急停按钮
- 控制柜上的急停按钮（只有 IRC-I8A-S 和 IRC-I6A-C 类型控制柜上有急停按钮）
- 外部急停装置（输入信号）

在出现危险情况或紧急情况时必须按下该装置。按下紧急停止装置时，机器人的反应如下：

机器人以断开停止（0 类）方式停机。

若要继续运行，必须先旋转紧急停止装置以将其解锁，再接通伺服电源。



连接在机器人上的工具或其他装置如可能引发危险，必须将其连入机器人的紧急停止回路中。如果没有遵照执行，则可能会造成作业人员死亡、严重的身体伤害或财产损失。

使能装置

捷勃特机器人的使能装置是位于机器人示教器反面的使能按钮，其具有三个位置：

- 未按下
- 中间位置
- 完全按下（警报位置）



只有当使能按钮保持在中间位置时，方可在手动模式下运行机械手臂。执行机器人示教操作或者手动模式运行程序过程中，松开使能按钮会触发 1 类停止方式；而完全按下使能按钮会触发 0 类停止方式。

外部安全接口

1. 外部紧急停止接口

每个可能控制启动机器人或引发可能带来危险情况的工位上都必须装有外部紧急停止装置，此装置由系统集成商来承担责任。为确保即使在控制 TP 已拔出的情况下也有紧急停止装置可供使用，至少要安装一个外部紧急停止装置。

外部紧急停止装置通过控制器上提供给客户的安全接口连接，外部紧急停止装置不包括在工业机器人的供货范围中。

2. 外部隔离性防护装置接口

为防止人员进入工业机器人的危险区域，系统集成商必须负责使用防护装置。该防护装置为隔离性防护装置，必须符合下列要求：

- 符合 EN ISO 14120 规范要求。
- 能够阻止人员进入危险区域并且不能被轻易越过。
- 本身不存在危险而且不会引发危险。
- 必须与所有危险位置保持规定的距离。

3. 外部限位停止装置接口

为防止机器人超出应用中的工作范围的装置。本身不存在危险而且不会引发危险。

1. 产品概要

本章节说明捷勃特机器人的基本构成和各类装置的概要。

1.1 机器人系统

机器人本体是指机械本体组成，机械本体由电机、减速机、编码器和传动机构等部分组成。

捷勃特机器人系统包含：

- 机器人本体
- 机器人控制柜
- 示教器
- 动力及编码器线缆

• 软件系统

捷勃特机器人系统如图所示：



1- 机器人本体



2-控制柜



3-示教器

图 1.1 四轴机器人系统组成图



1-机器人本体



2-控制柜



3-示教器

图 1.2 六轴机器人系统组成图

1.2 机器人型号说明

PUMA 系列											
产品类别		产品系列		负载		版本		臂展		分支版本	
GBT	捷勃特工业机器人	P	PUMA 6轴机器人	7	7KG	A	第一代	700	721mm	空白	标准版本
				20	20KG	B	第二代	900	901mm	C	洁净室版本
								1800	1805mm		

SCARA 系列													
[] — [] — [] — [] — [] — [] — []													
产品类别		产品系列		负载		版本		臂展		Z 轴行程		分支版本	
GBT	捷勃特工业机器人	S	SCARA 机器人	3	3KG	A	第一代	400	400mm	空白	标准行程	空白	标准版本
				6	6KG			500	500mm	.3	300mm	C	洁净室版本
				10	10KG			600	600mm				
				20	20KG			700	700mm				
								800	800mm				
								1000	1000mm				
机器人控制柜													
[] — [] — [] — [] — []													
产品类别		技术特征		标配轴数		版本		柜型					
IRC	工业机器人控制柜	I	集成一体式	4	4轴	A	第一代	空白	标准型				
		D	驱动分布式	6	6轴	B	第二代	S	小型				
				8	8轴			C	紧凑型				

1.3 控制装置

1.3.1 示教器

示教器是主管应用工具软件和用户之间接口的操作装置，用于手动控制、配置和编程工业机器人，使其执行特定任务。示教器通过电缆与控制装置连接。示教器外观如图 1.4 所示。



图 1.4 示教器正面及背面



示教器顶部 USB 口无法用于备份还原功能，请使用控制柜上的 USB 接口进行操作。

下表对示教器各按钮功能进行详细说明：

按键图标	功能
	示教器开关按钮：控制器上电后，长按此按钮启动示教器，看到该按钮橙色指示灯亮时松开按钮，机器人完成启动后，开关按钮指示灯由橙色变成绿色
	模式开关：旋转钥匙，可在手动限速模式（L）、手动全速模式(M)、自动模式（A）之间进行切换
	急停按钮:用于紧急情况下，制动机器人
	状态指示灯： <ul style="list-style-type: none"> • Fault 状态灯:机器人有报警信息时亮红色，无报警时不亮 • Auto 状态灯:机器人为自动模式时亮绿色，否则不亮 • Servo ON 状态灯:机器人在伺服上电状态亮绿色，否则不亮 • Runing 状态灯:机器人运行程序时绿灯闪烁
	按下此按钮进入当前使用的程序界面
	按下此按钮进入当前位置界面

	按下此按钮进入 I/O 监视界面
	按下此按钮进入数值寄存器界面
	按下此按钮进入位姿寄存器界面
	按下此按钮进入机器人负载设置界面
	按下此按钮返回示教器首页
	启动按钮：用于机器人程序的启动及调试
	暂停及停止按钮：在执行状况下，为暂停程序运行；在暂停状况下，为停止程序执行。
	点动键：用于点动进给；在没有报警的前提下，使能开关在中间档保持并且模式开关在手动模式下，按下点动键可移动机器人
	对事件或者错误进行确认并使机器人励磁
	自动模式下用于开启伺服
	自动模式下用于关闭伺服
	倍率键用来进行速度倍率的变更

使能开关	分未按下、中间档保持、完全按下三档，用于使机器人励磁 注：未按下与完全按下机器人都不会伺服上电，只有中间档保持才会伺服上电
------	--

1.3.2 控制柜面板

控制柜操作面板上有状态指示灯、控制柜开关、通信接口等，详细说明详见图 1.5 和 1.6。

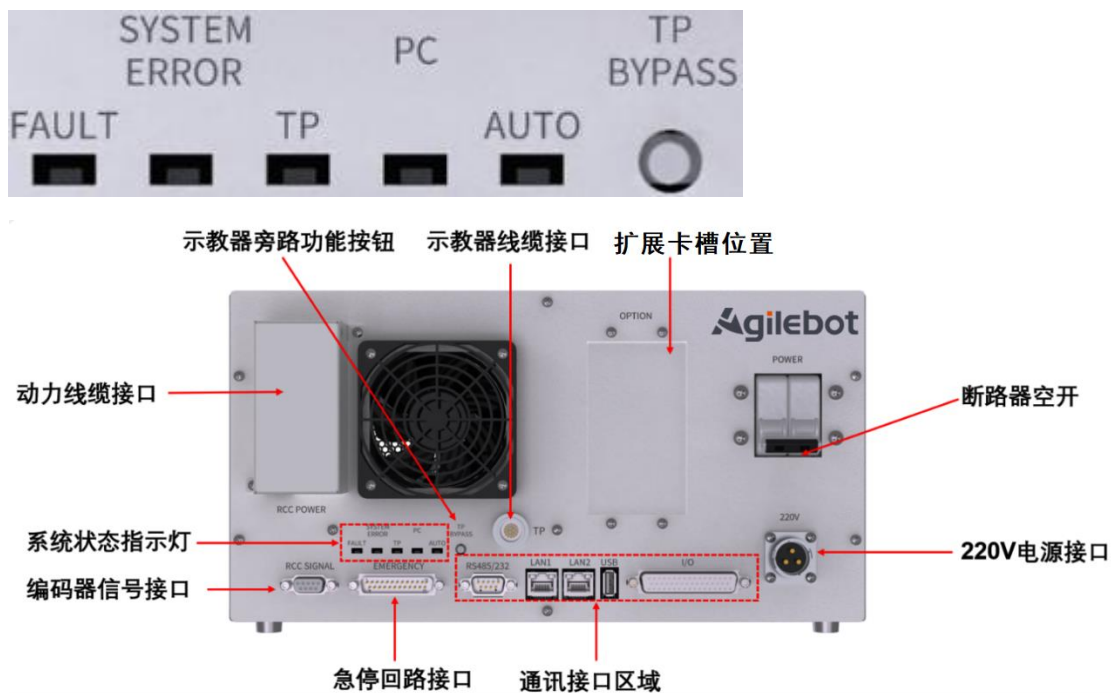


图 1.5 IRC-I4A-C 控制柜面板接口

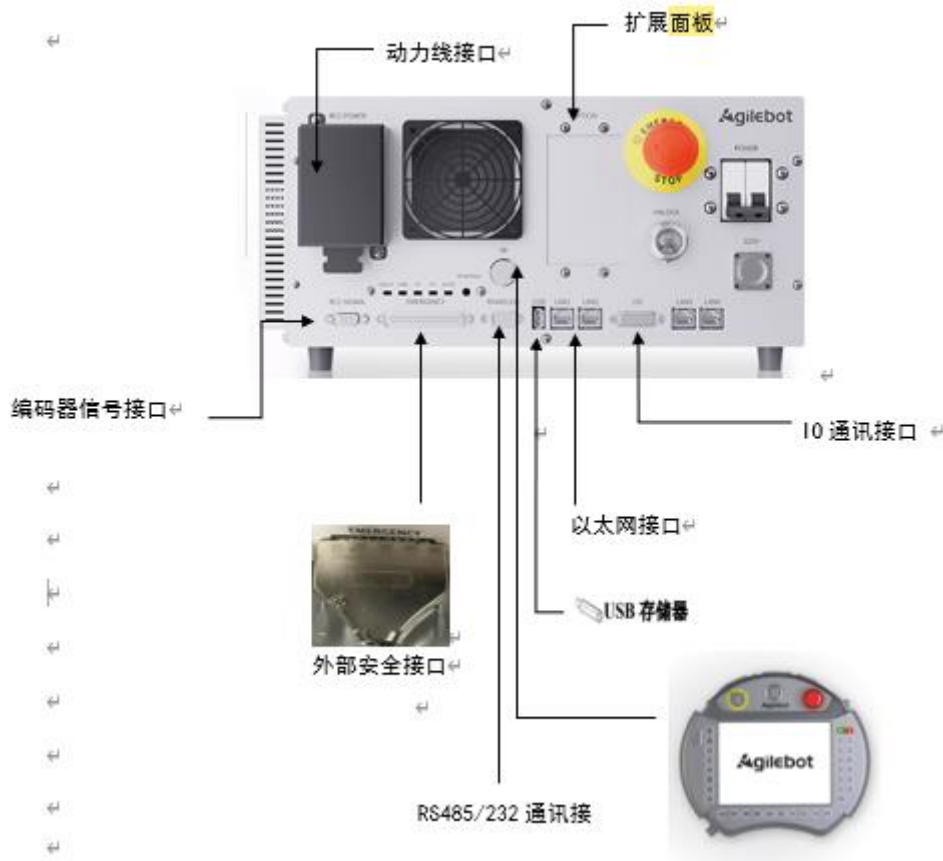


图 1.6 IRC-I6A-C 控制柜面板接口



除 TP（示教器）接口外，控制柜上所有接口均不支持热插拔，带电插拔可能导致设备损坏或引发安全风险！

1.3.3 IRC-I4A-C 控制柜不带示教器模式

IRC-I4A-C 系列控制柜不标配示教器，可以使用虚拟示教器软件 Compass 进行操作。

图 1.4 系统状态指示灯及 TP BYPASS 按钮介绍如下表所示

系统状态指示灯及按钮	功能
FAULT 状态灯	机器人报错亮红色，正常无报错不亮
SYSTEM ERROR 状态灯	当出现高等级的错误，即不可通过 rest 按钮或者 rest 信号清错复位的严重错误（一般就是 system

	等级), system error 指示灯亮起。
TP 状态灯	控制柜插入示教器时, 该灯亮绿色, 否则不亮
PC 状态灯	使用虚拟示教器软件 Compass 进行操作, 该灯亮绿色, 否则不亮
AUTO 状态灯	机器人的运行模式为自动运行时, 该灯亮绿色, 否则不亮
TP BYPASS 按钮	拔出示教器前, 若机器人是手动模式, 拔出示教器后可常按此按钮将机器人切换到自动模式

IRC-I4A-C 系列控制柜示教器支持热插拔, 拔出示教器时 **FAULT** 状态灯会亮红色, 通过外部复位或虚拟示教器软件 **Compass** 进行复位即可正常使用机器人。若 IRC-I4A-C 系列控制柜上插有示教器, 要使用 **Compass** 操作机器人, 必须将示教器切换到自动模式, **Compass** 才能够连接上控制柜, 并且拥有操作权限; **Compass** 拥有操作权限后, 示教器失去操作权限, 此时 PC 状态灯会亮绿色, TP 状态指示灯不亮; 若示教器为手动模式, 则 **Compass** 连接不上控制柜, 示教器具有操作权限。若 IRC-I4A-C 系列控制柜上没插示教器, 则可通过 **Compass** 直接连接 IRC-I4A-C 系列控制柜。

1.3.4 输入输出信号

I/O (输入/输出信号) 可使用通用信号和专用信号使机器人控制系统和外部设备进行数据的收发。通用信号 (用户定义的信号) 由程序进行控制, 进行与外部设备之间的通信。专用信号 (系统定义的信号) 由系统进行控制。

I/O 具有如下种类。

- 数字 I/O
- 组 I/O
- 专用 I/O
- 机器人 I/O

为了方便 IO 接线, 控制柜配备了外置的 I/O 转接板供客户使用。

1.3.5 机器人的动作

机器人的动作, 将从当前位置到目标位置的工具中心点的运动作为一个动作指令

来处理。

机器人控制装置使用综合控制机器人的轨迹、加减速、定位、速度的动作控制系统。

机器人控制装置，可以将多个轴分割为多个动作组进行控制（多动作功能）。各自的动作组相互独立，但是可以同步地使机器人同时动作。

机器人的动作有两类：来自示教器的点动进给和基于程序中的动作指令。

- 基于点动进给的机器人的动作，通过示教器的按键执行。点动进给时的动作，通过手动进给坐标系和速度倍率来确定。
- 基于动作指令的机器人的动作，通过动作指令中所指定的位置数据、动作类型、定位类型、移动速度、速度倍率等来确定。

动作类型有“MOVEJ”（关节）、“MOVEL”（直线）、“MOVEC”（圆弧）以及门型运动（具体见 3.3.1 节），可从中选择来操作机器人。选定“MOVEJ”时，工具中心点在两个示教点之间做非线性移动。选定“L”时，工具中心点在两个示教点之间作直线移动。选定“MOVEC”时，工具中心点从开始点经过过渡点到达目标点以圆弧方式对工具中心点移动进行控制的运动方式。工具中心点定位类型有两种：“FINE”（定位）和“SD”（转角半径）。

1.3.6 急停装置

机器人具有如下急停装置。

- 示教器右上的急停按钮
- 控制柜上的急停按钮（只有 IRC-I8A-S 和 IRC-I6A-C 类型控制柜上有急停按钮）
- 外部急停装置（输入信号）

按下急停按钮时，或者输入了外部急停信号时，机器人在任何情况下都会紧急停止。

1.4 接通电源和断开电源

本节内容主要说明如何启动/关闭机器人控制系统

1.4.1 开机前检查

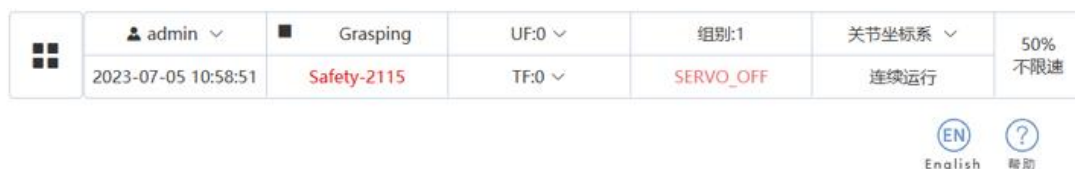
为确保机器人运行过程中机器人本身以及作业人员的安全，必须保证安装的正确

性；需要对机器人的安装进行如下步骤的检查：

- 机械检查：检查底座是否完全固定，检查末端工具安装是否牢固。
- 接线检查：检查动力线缆和编码器线缆接线是否正确及紧固。
- 控制柜检查：控制柜（包含电源切断开关）应安装在便于操作与维修的高度，建议的安装高度为 0.6m-1.9m 之间。

1.4.2 开机

开机顺序：连接控制柜电源线且断路器处于断开状态闭合断路器，控制柜通电 按示教器开机按钮，按钮黄灯亮起时松开等待开机。正常开机且自检通过后，显示画面如图 1.7 所示。



软件总版本: 6.0.0A (GBT-S6A-600)
 TP版本: 6.0.19_20230426.4e3e38e2c
 运控版本: 2.4.3.20230320.9a602223
 控制柜编号: 22031782010015A1
 本体编号: -1
 TP编号: AD210900021
 控制柜IP: 192.168.110.2
 PC访问路径: http://192.168.110.103:5613

图 1.7 示教器界面主页

1.4.3 关机

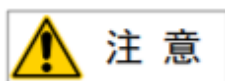
关机顺序：

1. 按下 TP 上的电源开关按钮直到出现如图 1.8 所示，点击确认。



图 1.8 关机提示窗

2. 当示教器开关电源指示灯由亮变为不亮时，说明示教器已经关机，此时断开控制柜断路器，即可完成关机。



必须等控制柜状态指示灯完全熄灭后才能重新开机。

1.5 操作界面



图 1.9 示教器界面主页

1.5.1 菜单

点击图 1.9 中左上角的“菜单按钮”，弹出菜单列表。详细的菜单列表及子列表如下：

菜单



图 1.10 菜单列表

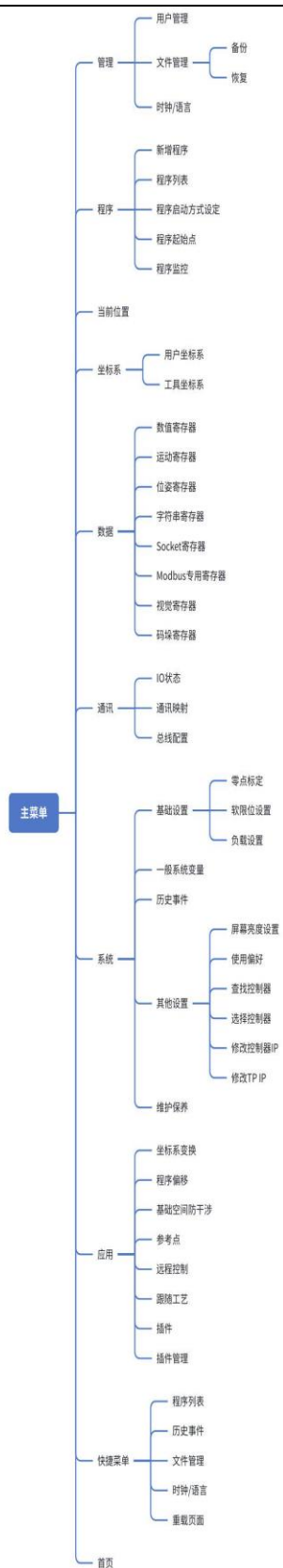


图 1.11 主菜单及子菜单列表

1.5.2 状态栏

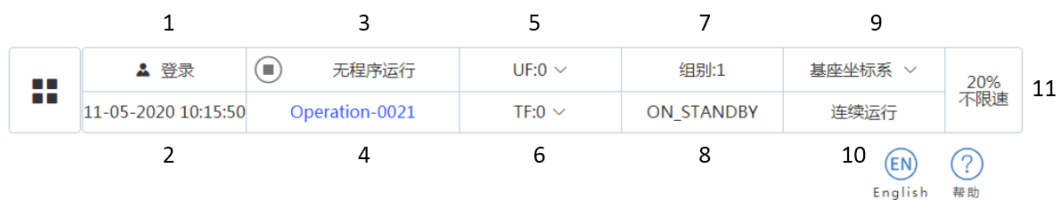


图 1.12 状态栏

下表详细说明状态栏中各组件具体含义：

状态栏编号及图标	具体含义
1	用户权限登录(具体权限定义见 1.1 节)
2	当前系统时间
3	显示当前运行程序名，程序名左侧标有程序当前状态（运行、暂停、停止状态），最右侧数值表示当前程序执行行号 <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;">  <p>：表示程序停止</p> </div> <div style="text-align: center;">  <p>：表示程序暂停</p> </div> <div style="text-align: center;">  <p>：表示程序运行</p> </div> </div>
4	报警事件（见第八章）
5	用户坐标系号切换/激活（见 1.7 节）
6	工具坐标系号切换/激活（见 1.7 节）

7	运动组显示, “1”表示机器人本体
8	伺服状态显示,“ON_STANDBY”表示伺服上电, “SERVO_OFF”表示伺服没上电
9	当前使用坐标系切换 (见 1.7 节)
10	显示程序运行方式 (连续运行、单步运行)
11	当前全局速度, 单击可进行调节 (点击后界面见图 1.21)
 帮助	用于查看机器人详细控制系统版本信息及常见问题 (见 1.5.3 节)
 English	示教器中、英、越、日、韩、俄文界面切换

1.5.3 系统信息

如图 1.9 在主界面右下角会显示控制系统当前简要的版本信息, 点击右上角帮助关于, 能够查看更详细的系统软件版本信息如图 1.13 所示。

← 返回首页
关于捷勃特工业机器人 Beta
版本二维码

名称	版本号
1 Robot Model	GBT-P7A-700
2 General Version	7.4.D.0
3 Controller	1.1.0.20241023.aa06593
4 TP-Charcoal	6.5.N.0(20241024.794127e2d)
5 File Service	6.5.N.0(20241024.794127e2d)
6 Hand Off	19628104
7 IO Board 0	20305144
8 RBF	21622161
9 Safety Board	21040917
10 Servo Control	JTAC-P7A-Debug-0.12.0-0, Servo-8.8PUMA20240716
11 Servo Param	P7A-700-20220901-00
12 U-Boot	21
13 TP SN	AD210500220
14 Controller SN	C010ABA210601B115
15 Body SN	-1

Copyright © 2020-2024 Agilebot Robotics Co., Ltd. - All Rights Reserved.
Open-source software

图 1.13 系统软件版本信息

1.6 模式开关

模式开关是安装在示教器上的钥匙操作开关，如图 1.14 所示。模式开关用于根据机器人的动作条件和使用情况选择最合适的机器人操作方式。操作方式分为 M (Manual) 手动全速模式、L (Limit manual) 手动限速模式和 A (Auto) 自动模式三种。

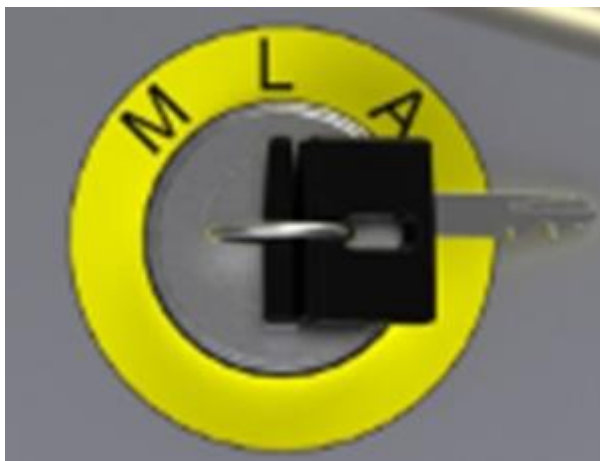


图 1.14 模式开关

要在没有运行程序的情况下切换操作模式。将钥匙从开关上拔出，即可将开关固定在该位置。

1. 手动全速模式 (M)

这是一种给机器人程序调试员或操作员（以下简称用户）调试机器人运动的模式。此模式下，用户主要被允许以下操作：

- 示教机器人
- 调试执行程序，包括正序连续执行程序、正序单步执行程序、逆序单步执行程序
- 编辑修改机器人程序

此模式下，用户主要被禁止以下操作

- 通过外部信号启动执行机器人程序

2. 手动限速模式 (L)

这是一种机器人运行速度受到限制的手动模式，目的和手动模式一样。只是在控制上会调整保持机器人的运动速度在 250mm/s 或者 18.5°/s 以下防止手动操作时速度过快而产生意外。

此模式下，用户主要被允许以下操作：

- 示教机器人
- 调试执行程序，包括正序连续执行程序、正序单步执行程序、逆序单步执行程序
- 编辑修改机器人程序

此模式下，用户主要被禁止以下操作：

- 通过外部信号启动执行机器人程序

此模式下，无论程序调试或执行时，对机器人运动速度做出以下限制：

- 笛卡尔运动指令运动始终低于 250mm/s 的速度
- 关节指令运动始终低于 18.5°/s
- 以倍率 100% 下的示教速度为基准进行速度限制。因此，在诸如示教速度为 2000mm/s 的情况下，若倍率为 100%，速度就被限制为 250mm/s，而倍率若为 50%，则速度被限制为 125mm/sec，如此通过降低倍率，就可进一步放慢速度。

3. 自动模式 (A)

这是用于机器人正常工作时自动运行的模式。该模式下机器人通过通讯或 IO 获取到要执行的程序信息或程序编号，并予以执行。

此模式下，用户主要被允许以下操作：

- 通过“程序启动模式”中选择的启动方式执行机器人程序

此模式下，用户主要被禁止以下操作

- 示教机器人
- 调试执行程序，包括正序单步执行程序、逆序单步执行程序
- 编辑修改机器人程序
- 修改机器人相关配置

1.7 示教机器人

按关节坐标系示教机器人（关于坐标系内容请参考 2.3 坐标系设置）。

机器人单轴操作可以单独操作机械臂每轴正方向或者反方向移动。具体操作过程如下：

1. 将机器人模式开关拨到 L 或 M 模式（手动模式）
2. 在示教器状态栏，选择机器人操作参考坐标系，如图 1.15 所示。



图 1.15 参考坐标系

3. 机器人操作参考坐标系选择的是用户坐标系或工具坐标系，如下图（若参考坐标系选择的不是用户坐标系或工具坐标系，则忽略该步骤）。

UF:1 ▾	TF:0 ▾
UF[0]	TF[0]
UF[1]	TF[1]
UF[2]	TF[2]
UF[3]	TF[3]
UF[4]	TF[4]
UF[5]	TF[5]
UF[6]	TF[6]
UF[7]	TF[7]
UF[8]	TF[8]
UF[9]	TF[9]
UF[10]	TF[10]

图 1.16 激活用户或工具坐标系

4. 将示教器反面使能(deadman)按钮，保持在中间档，使能按钮如图 1.17 所示。



图 1.17 示教器使能按钮

5. 按下示教器上的“Reset”按钮，清除报警，观察状态指示灯“SERVO ON”有没有亮绿灯，亮绿灯表示机器人伺服成功上电。
6. 按点动按钮进行示教机器人运动，点动按钮如下图 1.18 所示。

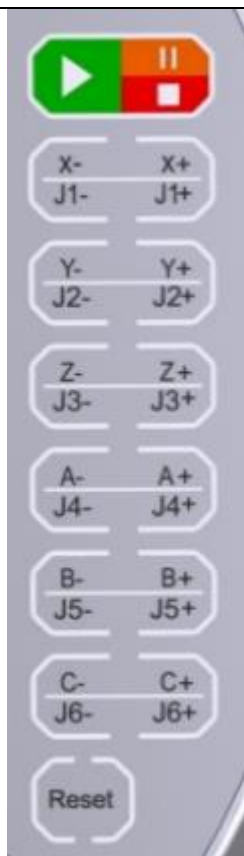


图 1.18 机器人点动按钮



在关节坐标系下示教时，点动按钮控制机器人的各轴旋转。

在笛卡尔坐标系下示教时，点动按钮 X,Y,Z 控制机器人工具坐标原点沿笛卡尔坐标系的 X,Y,Z 移动；点动按钮 A,B,C 控制机器人工具坐标原点绕笛卡尔坐标系的 X,Y,Z 转动。

使用世界坐标系或基座坐标系示教机器人

基座坐标系位于机器人底座，随出厂设定，世界坐标系定义默认与基座坐标系重合。示教时，所选择的工具坐标系或者法兰坐标系原点沿世界坐标系或者基座坐标系的方向移动或者旋转。

使用用户坐标系示教机器人

示教时，所选择的工具坐标系或者法兰坐标原点沿用户坐标系的方向移动或者旋转。

使用工具坐标系示教机器人

示教时，所选择的工具坐标系沿当前工具坐标系的方向移动或者旋转。工具坐标系是随其坐标系原点空间变化的坐标系，因此，每一次在示教动作都会改变工具坐标系的原点或者方向在空间中的位置与姿态。

使用 RTCP 用户坐标系/工具坐标系示教机器人

外部 TCP 下点动与用户坐标系下点动相似，均沿着参考坐标系的 X、Y、Z 方向运动。外部工具下点动姿态与用户坐标系下点动姿态区别：用户坐标系下点动姿态，TCP 点绕着用户坐标系旋转。外部工具下点动姿态，TCP 点绕着外部工具坐标系在旋转。

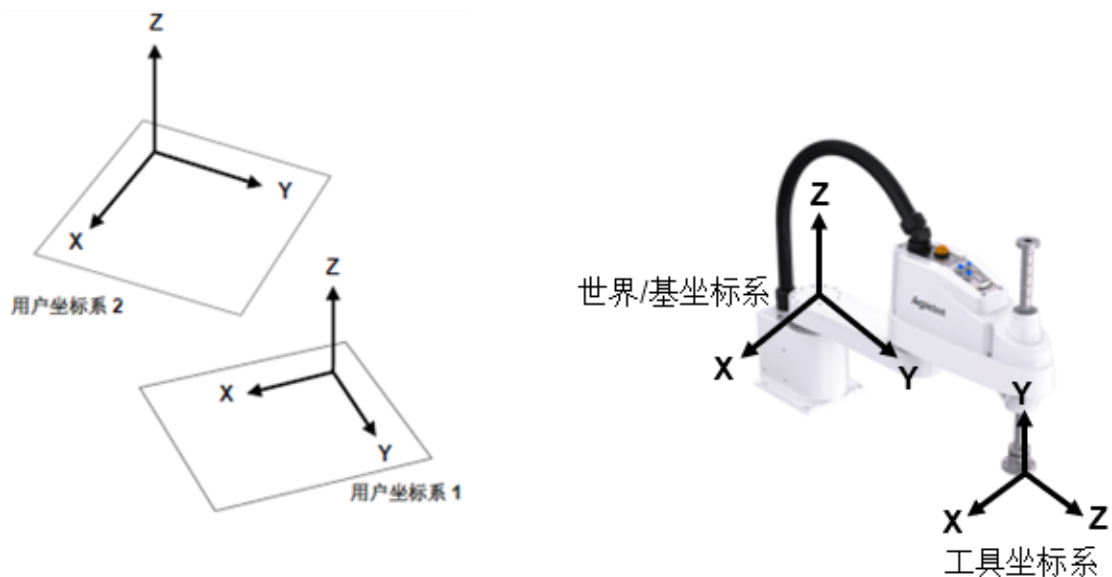


图 1.19 四轴机器人坐标系

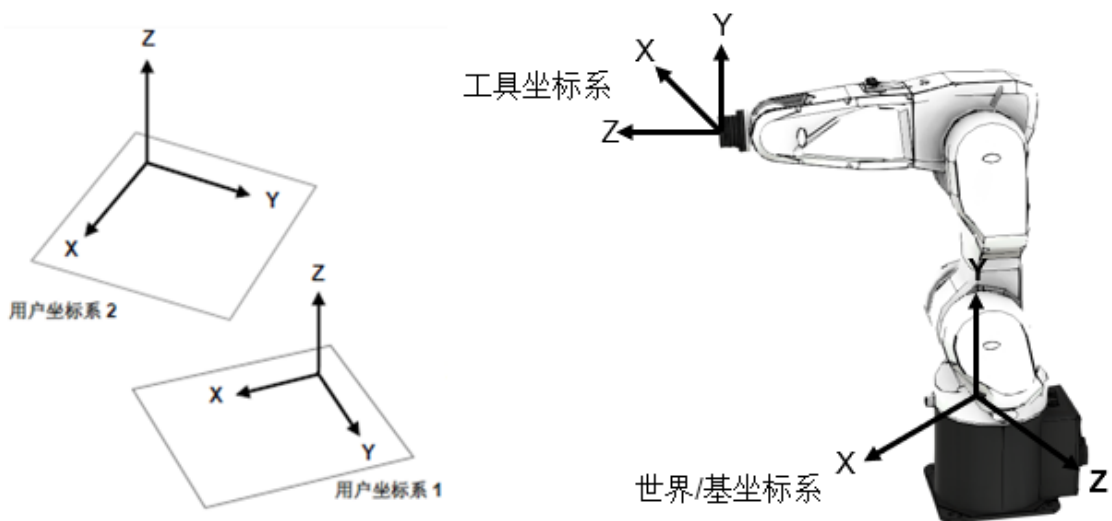


图 1.20 六轴机器人坐标系

1.8 速度控制

全局速度控制

全局速度控制功能适用于的操作模式：手动限速模式、手动模式、自动模式。

开机后，无论此时机器人模式开关处于什么模式，全局速度系数均为 10%。

用户可手动输入速度数值或拉动速度滑条来调节速度。

- 运行指令时：机器人速度限制值=指令速度*全局速度系数%（手动限速模式速度最大为 250mm/s or 18.5°/s）
- 示教时：机器人速度限制值=250mm/s*全局速度系数% 或者 18.5°/s*全局速度系数%

运动中的机器人，在调节该全局速度后，下一条运动指令解析执行时变更生效。

步进示教（使用进给量）

步进示教功能适用于：机器人手动（限速）模式、手动全速模式。自动模式不能进行进给运动控制功能。每当操作模式被切入自动模式，全局速度变为 10%。

步进示教功能启用后，每次按下机器人运动按钮机器人只移动相应步长的进给量。进给量设置的有效区间为(0, 10]mm、对应关节坐标运动步长和姿态角转动步长为(0, 5]°。若用户将其切换到自动模式，禁止该按钮点击。

注：步进运动规划时，给定的速度限制固定使用 250mm/s。

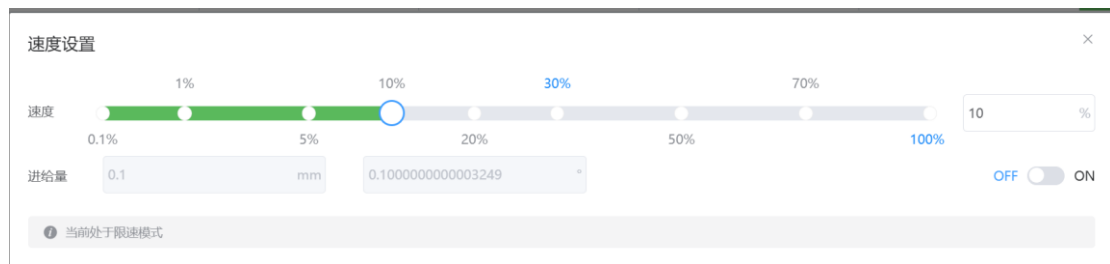


图 1.21 速度设置窗口

2. 机器人系统设定

2.1 I/O 信号

I/O（输入/输出信号），是机器人与末端执行器、外部装置等系统的外围设备进行通信的电信号。有通用 I/O 和专用 I/O。

通用 I/O

数字 I/O DI [i] / DO [i]

I/O 的 [i] 表示信号号码的逻辑号码。

专用 I/O

专用 I/O 是用途已经确定或机器人系统内可定义的 I/O，如 UI[1]至 UI[13]已确定用途。

系统 I/O UI [i] / UO [i]

I/O 的 [i] 表示信号号码的逻辑号码。

机器人 I/O

机器人 I/O 表示六轴机器人四轴手臂上的 IO 接口；四轴机器人没该信号接口。

机器人 I/O RI [i] / RO [i]

I/O 的 [i] 表示信号号码的逻辑号码。

I/O 映射

将通用 I/O（DI/O）和专用 I/O（UI/UO 等）称作逻辑信号。

相对于此，将实际的 I/O 信号点称作物理信号。要指定物理信号，利用 I/O 映射功能来指定 I/O 模块，并利用该 I/O 模块内的信号编号（物理编号）来指定各信号。

物理编号

物理编号系指 I/O 模块内的信号编号。按如下所示方式来表述物理编号。

- 数字输入信号: Input Port1、Input Port2 ...
- 数字输出信号: Output Port1、Output Port2 ...

为了在机器人控制装置上对 I/O 信号点进行控制，必须建立物理信号和逻辑信号的关联。将建立这一关联称作 I/O 映射（配置）。

• 有关数字 I/O、系统 I/O，可变更 I/O 分配，重新定义物理信号和逻辑信号的关联。

DI/DO 转接板卡

为了使 I/O 信号接线及使用方便，适配了 IO 转接板卡。

IRC-I4A-C 控制柜适配转接板卡的接线方式应适配控制柜的类型，NPN 型的控制柜转接板卡就按 NPN 的方式接线，PNP 型的控制柜转接板卡就按 PNP 的方式接线。IRC-I4A-C 控制柜有输出 16 点，输入 24 点的控制口；通过专用信号线缆与转接板卡连接。IRC-I4A-C 适配的转接板卡外观如下图所示。



图 2.1 外置 IO 板

下表详细说明了 IO 板各个端口默认线序定义：（实际功能可根据使用场景设定）

端口序号	功能/物理编号	端口序号	功能/物理编号
01	DI 1-8 公共端	26	Input Port16
02	Input Port1	27	Output Port6
03	Input Port2	28	Output Port7
04	Input Port3	29	Output Port8
05	Input Port4	30	Output Port9
06	Input Port5	31	Output Port10
07	Input Port6	32	Output Port11
08	Input Port7	33	NC
09	Input Port8	34	DI 17-24 公共端

10	Output Port1	35	Input Port17
11	Output Port2	36	Input Port18
12	Output Port3	37	Input Port19
13	Output Port4	38	Input Port20
14	Output Port5	39	Input Port21
15	DO_PS_IN 1	40	Input Port22
16	DO 1-8 公共端	41	Input Port23
17	24V	42	Input Port24
18	DI 9-16 公共端	43	Output Port12
19	Input Port9	44	Output Port13
20	Input Port10	45	Output Port14
21	Input Port11	46	Output Port15
22	Input Port12	47	Output Port16
23	Input Port13	48	DO_PS_IN 2
24	Input Port14	49	DO 9-16 公共端
25	Input Port15	50	0V

IR 控制柜 IRC-16A-C 针对 I/O 板卡的 48 路 DI/D0 专门开发了相应的 I/O 转接板卡辅助客户使用，I/O 转接板卡如下图所示：

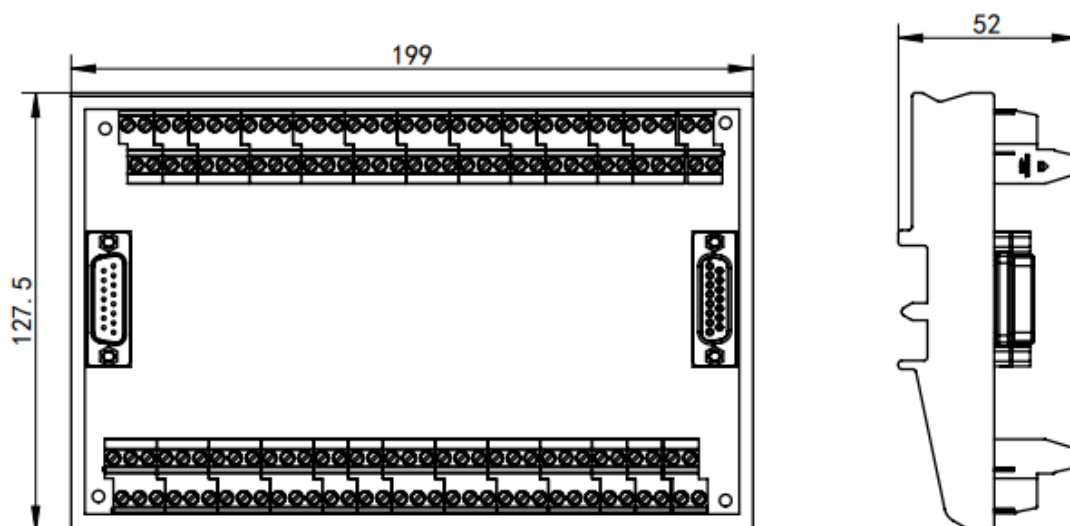
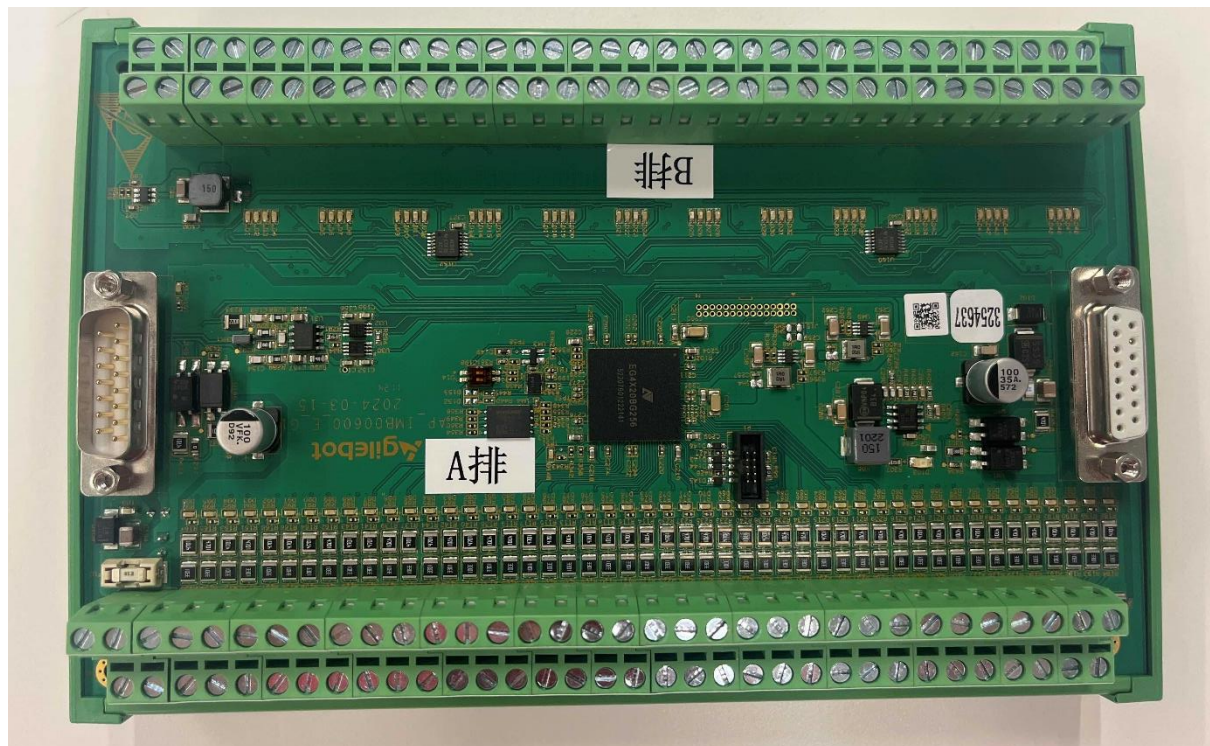


图 2.2 外置 IO 板

I/O 转接板共有 4 组端子排，其中 A 排端子排为 DI 接口，B 排端子排为 D0 接口，另外，还有一组 24V 和 0V 的端子排，各 4PIN。

I/O 板卡 IMB 总计提供 48 路 DI 和 48 路 D0。

有关详细的设定操作详见示教器的操作手册，本说明书仅对 DI/DO 的电路连接做出说明。

A 排的引脚定义如下表所示：（实际功能可根据使用场景设定）的引脚定义如下表所示。

端口序号	信号名称	端口序号	信号名称
01	Input Port1	33	Input Port25
02	Input Port2	34	Input Port26
03	Input Port3	35	Input Port27
04	Input Port4	36	Input Port28
05	Input Port5	37	Input Port29
06	Input Port6	38	Input Port30
07	Input Port7	39	Input Port31
08	Input Port8	40	Input Port32
09	Input Port9	41	Input Port33
10	Input Port10	42	Input Port34
11	Input Port11	43	Input Port35
12	Input Port12	44	Input Port36
13	COM_1_12	45	COM_25_36
14	24V	46	24V
15	0V	47	0V
16	0V	48	0V
17	Input Port13	49	Input Port37
18	Input Port14	50	Input Port38
19	Input Port15	51	Input Port39
20	Input Port16	52	Input Port40
21	Input Port17	53	Input Port41
22	Input Port18	54	Input Port42
23	Input Port19	55	Input Port43
24	Input Port20	56	Input Port44

端口序号	信号名称	端口序号	信号名称
25	Input Port21	57	Input Port45
26	Input Port22	58	Input Port46
27	Input Port23	59	Input Port47
28	Input Port24	60	Input Port48
29	COM_13_24	61	COM_37_48
30	24V	62	24V
31	0V	63	0V
32	0V	64	0V

DI 接口——A 排引脚定义表

B 排的引脚定义如表所示。

端口序号	信号名称	端口序号	信号名称
01	0V	33	0V
02	0V	34	0V
03	24V	35	24V
04	24V_In_1-12	36	24V_In_25-36
05	Output Port1	37	Output Port25
06	Output Port2	38	Output Port26
07	Output Port3	39	Output Port27
08	Output Port4	40	Output Port28
09	Output Port5	41	Output Port29
10	Output Port6	42	Output Port30
11	Output Port7	43	Output Port31
12	Output Port8	44	Output Port32
13	Output Port9	45	Output Port33
14	Output Port10	46	Output Port34
15	Output Port11	47	Output Port35
16	Output Port12	48	Output Port36
17	0V	49	0V

18	0V	50	0V
19	24V	51	24V
20	24V_In_13-24	52	24V_In_37-48
21	Output Port13	53	Output Port37
22	Output Port14	54	Output Port38
23	Output Port15	55	Output Port39
24	Output Port16	56	Output Port40
25	Output Port17	57	Output Port41
26	Output Port18	58	Output Port42
27	Output Port19	59	Output Port43
28	Output Port20	60	Output Port44
29	Output Port21	61	Output Port45
30	Output Port22	62	Output Port46
31	Output Port23	63	Output Port47
32	Output Port24	64	Output Port48

表 D0 接口——B 排引脚定义表

2.1.1 数字 I/O

数字 I/O (DI / DO), 是从外围设备通过处理 I/O 单元的输入/输出信号线来进行数据交换的标准数字信号。正确地说其属于通用数字信号。数字信号的值有 ON (通) 和 OFF (断) 共两类。

I/O 的映射 (配置)

数字 I/O 可对信号线的物理号码进行再定义。设定下列项目。有关 I/O 分配的详情, 请参照“2.1 I/O 信号”

起始端口为进行信号点的映射而将物理号码分配给逻辑号码。指定该分配的最初的物理号码。



物理号码指定 I/O 模块上的输入/输出引脚。逻辑号码被分配给该物理号码, 所以可以 1 个信号为单元改变分配。

物理号码的起始端口根据实际需要设定。

数字 I/O 的分配操作

依次点击“菜单按钮”→“通讯”→“IO 映射”进入图 2.3 所示界面

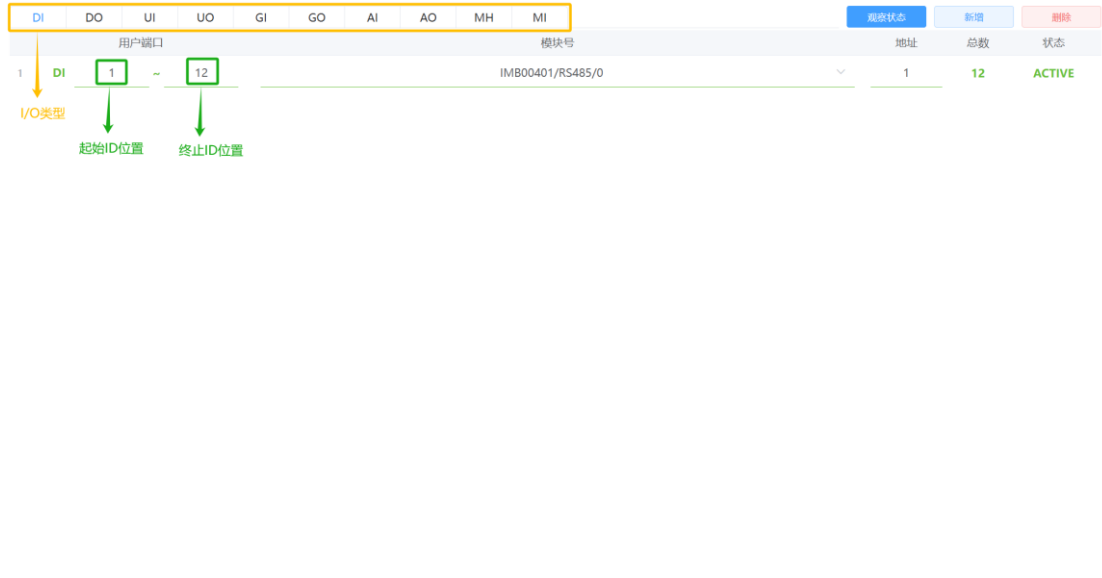


图 2.3 IO 映射界面

IO 映射界面的具体含义如下：

- IO 类型：用户可以根据所需管理与映射的 IO 类型来进行选择。目前可映射 IO 有 DI/DO/UI/UO；
- 输入：根据用户选择“IO 类型”不同，显示 UO/DO 或 UI/DI，实例图中为 DI。可在此选择要做 IO 映射的逻辑信号 ID；（方法：整段设置，即仅需输入需要设置的逻辑 IO 的起始 ID 和终止 ID。例如，需要设置 12 个 DI，只需要在起始 ID 位置输入 1 并在终止 ID 位置输入 12 即可）（注：同一类型 IO 的 ID 段落不能出现重叠）
- 模块号：此处允许用户选择当前系统中识别到的通信模块。如下图所示

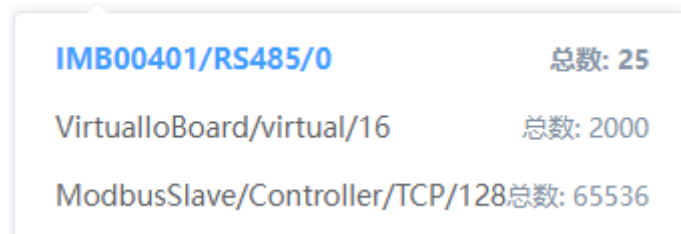


图 2.4 模块选择界面

- 模块 1 是机器人控制柜适配 IO 转接板卡
- 模块 2 是虚拟设备 IO，供用户在没有实际物理 IO 模块的情况下进行 IO 映射以便于调试程序

- 模块 3 是 modbus tcp 使用模块，可将逻辑信号映射到 modbus 寄存器中进行使用；DI 映射到 modbus 寄存器中的 Coil Status,DO 映射到 Discrete inputs.



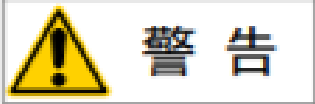
模块号下拉列表中仅有当前系统中识别到的通信模块以及 Virtual IO Device 供用户选择。Virtual IO Device 为虚拟 IO 设备 (2000 In/2000 Out)，供用户在没有实际物理 IO 模块的情况下进行 IO 映射以便于调试程序。

- 起始端口：此处是输入需要配置的逻辑 IO 段的第一个逻辑 IO 对应到 IO 模块上的物理端口。例：需要配 IO 段是 DI1-DI5，若用户选择对应的物理端口是 3，则对应关系是 DI1-DI5 与物理端口 Input Port3 - Input Port7 一一对应，即物理端口 Input Port3 有信号输入，则 DI1 状态为 ON；DI2-DI5 的状态依次类推。
- 总数：此处是根据起始 ID 与终止 ID 自动计算出来的 IO 个数 (个数=终止数-起始数+1)；(如 DI2-DI5，对应的 IO 个数为 5-2+1=4 个)。只读，用户不能操作。
- 状态：显示该 IO 映射每个条目的状态；分别为 Active、Unfindable、Modifying、Invalid 四种。
- Active 代表该 IO 配置处于激活状态 (配置成功并保存过的)。
- Unfindable 代表无法找到该物理模块 (此种情形特指原来是激活状态的，由于物理模块连接断开导致)。
- Modifying/To be Saved 代表修改过的 IO 配置，且该配置合法但还未进行保存。
- Invalid 代表该条目的配置内容无效，存在错误或地址冲突空缺项 (不合法)，用户无法成功保存该条目。
- 添加：单击该按钮，新建一个 IO 配置条目。
- 保存：单击保存映射的内容。
- 删除：单击删除当前所选的 IO 配置条目。
- 显示错误：显示当前错误，根据错误提示内容进行修改。
- IO 状态：点击此处可切换到 IO 状态界面

执行输出

可通过程序的执行或者手动操作设定机器人输出的值。

有关信号的强制输出、仿真输入 (见 2.2 节)。



控制装置通过信号进行外围设备的控制。强制输出 / 仿真输入，在某些情况下恐会给系统的安全性带来不良影响。在确认信号的使用方法及功能之前，请勿执行强制输出或仿真输入。

2.1.2 组 I/O

组 IO(GIGO)，是用来汇总多条信号线并进行数据交换的通用数字信号。组信号的值用数值(10 进制数)来表达，转变或逆转变为 2 进制数后通过信号线交换数据。

I/O 的映射（配置）

组 IO 可以将信号号码作为 1 个组进行定义。可以将 2~16 个信号作为 1 组进行定义。

2.1.3 专用 I/O

系统 I/O (UI/UO)，是在系统中已经确定了其用途的专用信号，即系统输入输出信号。这些信号从 IO 模块与远程控装置和外围设备连接，从外部进行机器人控制。UI 信号是外界可写，机器人只读的信号。UO 信号是外界只读，机器人可写的信号。关于专用 I/O 的映射参考 2.1.1 节 IO 映射内容。

默认提供的映射只有 UI[1]~UI[13]、UO[1]~UO[13]，其他功能默认不启用，以下是专用 IO 即系统输入输出的说明：

UI[1]	Servo_Ena ble 伺服使能信号（可用做瞬时停止的外围软件报警信号；或者在暂停后，关伺服上抱闸做完	Servo_Enable 通常情况为 ON，外围上位机不希望机器人运动或上电时输入 OFF，用做安全锁定。 为 OFF 时系统做如下处理： 1.发出警报后断	UO[1]	CMD ENBL E 允许 外围 设备 控制 机器 人的 状态	为 ON 表示允许外围设备控制，为 OFF 时不允许外围设备控制。 下列条件成立时输出高电平： 1.UI[5]为 ON 2.UI[4]为 OFF 2.模式开关处于
-------	---	--	-------	--	---

	全停止)	<p>开伺服电源</p> <p>2.瞬时停止机器人 (0 类停止), 暂停程序的执行。</p> <p>3.伺服始终无法上使能。</p> <p>旁路为 ON。</p>		信号。	<p>“Auto”模式</p> <p>3.UO[3]为 OFF</p>
UI[2]	<p>Pause_Request</p> <p>暂停信号</p>	<p>暂停信号, 通常为 ON, 为 OFF 时系统做如下处理:</p> <p>规划减速并减速停止执行中的动作, 暂停程序的执行。</p> <p>旁路为 ON。</p>	UO[2]	Paused	<p>“暂停中”状态信号。当程序执行状态处于“暂停状态 Paused”时, 该信号为 ON (即机器人暂停)。</p>
UI[3]	<p>Reset</p> <p>报警复位信号</p>	<p>报警复位信号, 解除报警, 伺服上电, 高电平有效产生 Reset 请求。</p>	UO[3]	FAULT报警信号	<p>报警信号在系统中发生报警时产生输出, 可以通过 RESET 来进行复位。</p> <p>注: 当系统发出警告类报警时 (Warning), 该信号不输出。</p>
UI[4]	<p>Start&Restart</p> <p>程序启动/恢复信号</p>	<p>启动或重启程序 (视当时的程序状态为 aborted 还是 Pause 决定), 其作用和 TP 上的启动按钮一样。取下降沿</p>	UO[4]	Program Running 程序正在运行中信号	<p>为 ON 时, 表示正在运行程序中; 为 OFF 时, 表示没有程序在运行</p>

		有效启动或重启程序。		号	
UI[5]	Abort Program 程序终止请求信号	<p>请求将执行中或暂停状态的程序终止。</p> <p>通常为 ON, 为 OFF 时系统做如下处理:</p> <ol style="list-style-type: none"> 1. 报警栏提示有程序终止请求, 程序进入终止模式。如果程序还在运行则先瞬时停止机器人的动作再终止程序, 类似“aborted”报警。 2. 允许伺服上使能和示教, 但无法手动或自动执行程序。 <p>旁路为 ON。</p>	UO[5]	Servo Status 信号	<p>机器人运行状态为“Working”、“On-Standby”、“Servo-ON”时, 该信号置高电平。</p> <p>“Servo-OFF”时置低电平。</p>
UI[6]	Selection Strobe 触发信号	<p>仅在“程序启动方式”设置为“主程序号启动”或“主程序号启动简易模式”时有效。</p> <p>读取选择要执行的程序的触发信号, 当其为 ON 时, 读取 Program Selection 1-6 的</p>	UO[6]	Selection Check Request	<p>仅在“程序启动方式”设置为主程序号启动”或“主程序号启动简易模式”时有效。</p>

		输入，选择要执行的程序。 注：当有程序正在执行（运行状态或暂停状态），忽略该信号。			
UI[7]	MPLCS Start	仅在“程序启动方式”设置为“主程序号启动”或“主程序号启动简易模式”时有效。 程序号选择的启动信号	UO[7]	MPLCS Start Done	仅在“程序启动方式”设置为“主程序号启动”或“主程序号启动简易模式”时有效。
UI[8]-UI[13]	Program Selection 1-6	仅在“程序启动方式”设置为“主程序号启动”或“主程序号启动简易模式”时有效。 程序号的 6 位二进制数，将其转换为的十进制数，即为需要启动的主程序的启动号。	UO[8]-UO[13]	Selection Confirm 1-6	仅在“程序启动方式”设置为“主程序号启动”或“主程序号启动简易模式”时有效。 在接收到 Selection Strobe 信号后，机器人控制器会去读 UI[8]-UI[13]的状态，并将其反馈给上位机确认。

2.1.4 机器人 I/O

机器人 I/O，是经由机器人，作为末端执行器 I/O 被使用的机器人数字信号。末端执行器 I/O 与机器人的手腕上所附带的连接器连接后使用。

末端执行器 I/O (机器人本体上的 IO 接口)由 6 个输入、6 个输出的通用信号构成。

RI [1~6] 输入

RO [1~6] 输出

本体上的 IO 接口在四轴手臂上,有两个接口, 分别是 EE1 与 EE2

引脚序号	功能
01	RO1
02	RO2
03	RO3
04	RO4
05	RI1
06	RI2
07	IO_24V
08	IO_0V
09	/

EE1 航插引脚定义

引脚序号	功能
01	RO5
02	RO6
03	RI3
04	RI4
05	RI5
06	RI6

07	IO_24V
08	IO_0V
09	/

EE2 航插引脚定义



四轴机器人本体上没有 IO 接口，即四轴机器人没有 RI/RO

2.1.5 手动控制 I/O

I/O 的手动控制，在执行程序前，相对外围设备进行信号的交互。

I/O 的手动控制是指如下条目。

- 强制输出
- 仿真输入

依次点击“菜单按钮”→“通讯”→“IO 状态”进入 IO 状态界面，如下图所示

DI/DO				IO类型			取消所有仿真		IO映射	
端口	名称	仿真	值	端口	名称	值				
DI[1]	<input type="text"/>	UnSim Sim	OFF	DO[1]	<input type="text"/>	OFF				
DI[2]	<input type="text"/>	UnSim Sim	OFF	DO[2]	<input type="text"/>	OFF				
DI[3]	<input type="text"/>	UnSim Sim	OFF	DO[3]	<input type="text"/>	OFF				
DI[4]	<input type="text"/>	UnSim Sim	OFF	DO[4]	<input type="text"/>	OFF				
DI[5]	<input type="text"/>	UnSim Sim	OFF	DO[5]	<input type="text"/>	OFF				
DI[6]	<input type="text"/>	UnSim Sim	OFF	DO[6]	<input type="text"/>	OFF				
DI[7]	<input type="text"/>	UnSim Sim	OFF	DO[7]	<input type="text"/>	OFF				
DI[8]	<input type="text"/>	UnSim Sim	OFF	DO[8]	<input type="text"/>	OFF				
DI[9]	<input type="text"/>	UnSim Sim	OFF	DO[9]	<input type="text"/>	OFF				
DI[10]	<input type="text"/>	UnSim Sim	OFF	DO[10]	<input type="text"/>	OFF				

共 1024 条 < > 前往 1 页

图 2.5 IO 状态界面

在该界面不仅可以查看 IO 的状态，还可以对数字 IO 进行输出信号的强制与输入信号的仿真，在上图界面“名称”条框中可以写入信号的名称（支持中文）

选择手动控制 IO 类型

点击 IO 类型可选择需要操作的 IO 类型，如下图所示

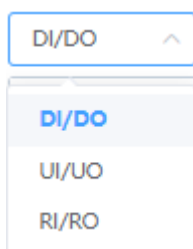


图 2.6 IO 类型切换窗口

强制输出

强制输出，将数字输出信号手动切换到 ON / OFF。

在 IO 状态界面点击下图绿框中的标识，会弹出 NO/OFF，此时选择需要强制的状态即可完成强制输出

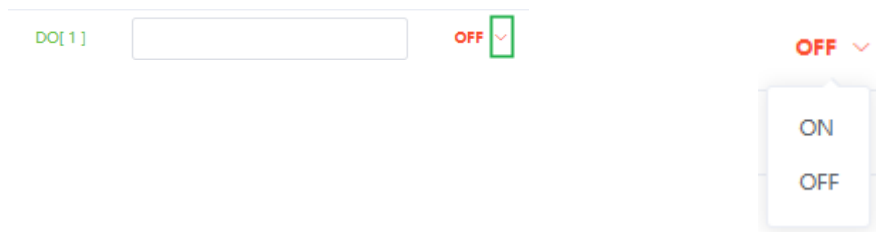


图 2.7 DO 强制输出窗口

输入信号仿真

为了实际调试方便，提供了输入信号仿真功能。点击“sim”即可开启仿真功能，点击“UnSim”关闭仿真功能。如下图所示，可对开启仿真功能的输入信号进行 ON/OFF 操作；点击“取消所有仿真”按钮，可取消所有仿真的信号。

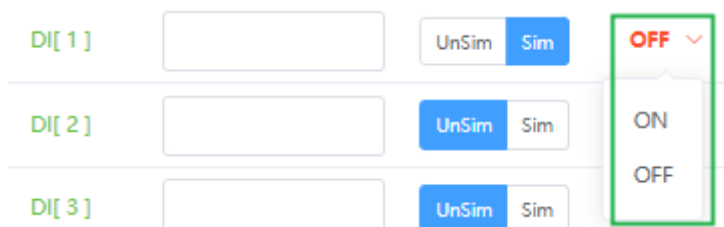


图 2.8 DI 仿真窗口

专用信号 UI 的旁路功能

UI 拥有旁路功能，在旁路功能未激活时，UI 的值显示其真实值（只可对 UI[1]/UI[2]/UI[5]进行旁路功能的操作）。

UI/UO				IO 映射		
端口	名称	旁路	值	端口	名称	值
UI[1]	Servo_Enable	<input checked="" type="checkbox"/> Yes <input type="checkbox"/> No	ON	UO[1]	CMD_Enable	OFF
UI[2]	Pause_Request	<input checked="" type="checkbox"/> Yes <input type="checkbox"/> No	ON	UO[2]	Paused	OFF
UI[3]	Reset	<input type="checkbox"/> Yes <input type="checkbox"/> No	OFF	UO[3]	Fault	OFF
UI[4]	Start&Restart	<input type="checkbox"/> Yes <input type="checkbox"/> No	OFF	UO[4]	Program_Running	OFF
UI[5]	Abort_Program	<input checked="" type="checkbox"/> Yes <input type="checkbox"/> No	ON	UO[5]	Servo_Status	OFF
UI[6]	Selection_Strobe	<input type="checkbox"/> Yes <input type="checkbox"/> No	OFF	UO[6]	Selection_Check_Request	OFF
UI[7]	MPLCS_Start	<input type="checkbox"/> Yes <input type="checkbox"/> No	OFF	UO[7]	MPLCS_Start_Done	OFF
UI[8]	Program_Selection_1	<input type="checkbox"/> Yes <input type="checkbox"/> No	OFF	UO[8]	Selection_Confirm_1	OFF
UI[9]	Program_Selection_2	<input type="checkbox"/> Yes <input type="checkbox"/> No	OFF	UO[9]	Selection_Confirm_2	OFF
UI[10]	Program_Selection_3	<input type="checkbox"/> Yes <input type="checkbox"/> No	OFF	UO[10]	Selection_Confirm_3	OFF

共 13 条 < > 前往 1 页

图 2.9 系统信号窗口

2.2 坐标系设置

坐标系是为确定机器人的位置和姿势而在机器人或空间上进行定义的位置指标系统。机器人具有如下多个坐标系系统。



图 2.10 机器人坐标系示意图

关节坐标系

关节坐标系是设定在机器人的关节中的坐标系。关节坐标系中的机器人的位置和姿势，以各关节的底座侧的关节坐标系为基准而确定。

直角坐标系

直角坐标系中的机器人的位置和姿势，通过从空间上的直角坐标系原点到工具侧的直角坐标系原点（工具中心点）的坐标值 X、Y、Z 和空间上的直角坐标系相对 X 轴、Y 轴、Z 轴周围的工具侧的直角坐标系的回转角 A、B、C 予以定义。

工具坐标系

这是用来定义工具中心点（TCP）的位置和工具姿势的坐标系。工具坐标系必须事先进行设定。未定义时，默认是末端法兰坐标系。

世界坐标系

被固定在空间上的标准直角坐标系，与基坐标系重合。

基座坐标系

被固定在机器人底座的标准直角坐标系，用户坐标系基于该坐标系而设定。

用户坐标系

用户坐标系，是用户对每个作业空间进行定义的直角坐标系。未定义时，将由世界/基坐标系来替代该坐标系。

2.2.1 设定工具坐标系

工具坐标系，是表示工具中心点（TCP）和工具姿势的直角坐标系。工具坐标系通常以 TCP 为原点，将工具方向取为 Z 轴。

工具坐标系，由工具中心点（TCP）的位置 (X,Y,Z) 和工具的姿态(X,Y,Z 轴的旋转角度)(A,B,C) 构成。

工具中心点（TCP）的位置，通过相对于默认坐标系的工具中心点的坐标值 x、y、z 来定义。

工具的姿势，工具坐标系相对于末端法兰坐标系的姿态来定义。

工具坐标系，在坐标系设定画面上进行定义，可定义 10 个工具坐标系，并可根据情况进行切换。

设定工具坐标系的优点：

- 可使工具沿工具坐标系方向上直线移动。
- 工具可绕工具坐标系旋转，保持工具工作点位置不变方向变化，从而改变机器人的姿态。
- 使程序设定的速度为工具工作点实际的速度，而不是默认法兰末端的速度。
- 默认工具坐标系就是末端法兰坐标系

设定方法

- 四点法
- 七点法（只适用于六轴机器人）
- 直接输入法



设置工具坐标系前，要把当前使用坐标系切换到基坐标系，在基坐标系下进行示教点位。

工具坐标系界面介绍

工具坐标系		用户坐标系
ID 1	Group ID 1	名称
备注		
* X	-4001.82 mm	* Y 1630.501 mm
* Z	1559.442 mm	
* A	-132.544 °	* B 22.967 °
* C	-175.024 °	

图 2.11 工具坐标系设定窗口

- 用户坐标系：点击会切换到用户坐标系设置界面
- ID：号表示当前选择的工具坐标系编号
- Group ID：表示运动组，目前支持机器人本体，本体 Group ID 用 1 表示
- 名称：可输入工具坐标系名称
- 备注：可对工具坐标系进行备注

六轴机器人七点法设置工具坐标系

七点法设置工具坐标系点位如图 2.12 所示；其中 P1-P5 的控制点要始终保持与校枪器的尖点接触，P1-P4 四个点的姿态尽可能差别大；示教 P5 点时，工具末端必须与校枪器保持在一条直线上；P6 点用来确定工具坐标 X 的方向，P7 点用来确定工具坐标 Z 的方向，示教时 P7 在 P6 的基础上向工具坐标 Z 方向移动一段距离即可。

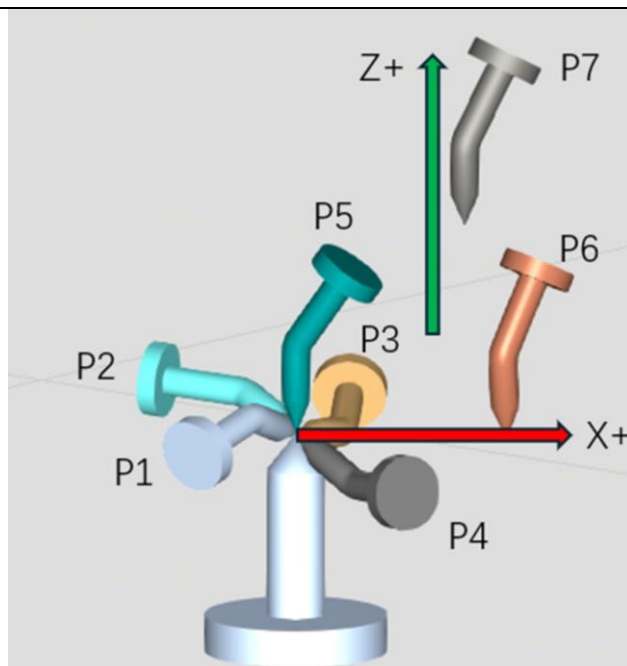


图 2.12 七点法设置工具坐标系点位

具体步骤：

- 1.依次点击“菜单按钮”→“坐标系”→“工具坐标系”进入工具坐标系设置界面如图 2.13 所示

用户坐标系

	工具坐标系		
TF[1]	ID 1	Group ID 1	名称 <input style="width: 80px;" type="text"/>
TF[2]	备注 <input style="width: 90%; height: 20px;" type="text"/>		
	*X <input style="width: 40px;" type="text" value="0"/> mm	*Y <input style="width: 40px;" type="text" value="0"/> mm	*Z <input style="width: 40px;" type="text" value="0"/> mm
	*A <input style="width: 40px;" type="text" value="0"/> °	*B <input style="width: 40px;" type="text" value="0"/> °	*C <input style="width: 40px;" type="text" value="0"/> °
	编辑		
删除 新建			

图 2.13 工具坐标系设置界面 1

2. 点击“新建”或者选择需要设置的工具坐标→点击“编辑”后界面如图 2.14 所示
 点击此处选择 7 点法后界面如图 2.15 所示

图 2.14 工具坐标系设置界面 2

工具坐标系 用户坐标系

ID 1 Group ID 1 ▾ 名称

备注

* X	<input type="text" value="-4001.82"/>	<input type="text" value="mm"/>	* Y	<input type="text" value="1630.501"/>	<input type="text" value="mm"/>	* Z	<input type="text" value="1559.442"/>	<input type="text" value="mm"/>
* A	<input type="text" value="-132.544"/>	<input type="text" value="°"/>	* B	<input type="text" value="22.967"/>	<input type="text" value="°"/>	* C	<input type="text" value="-175.024"/>	<input type="text" value="°"/>

方法 7p ▾

<input type="text" value="点位 1"/> *	<input type="text" value="点位 2"/> *	<input type="text" value="点位 3"/> *	<input type="text" value="点位 4"/> *
<input type="text" value="坐标原点"/> *	<input type="text" value="X方向"/> *	<input type="text" value="Z方向"/> *	

图 2.15 工具坐标系设置界面 3

3.选中需要示教的点位，在该点位示教完成后→点击“示教记录”后界面如图 2.16 所示

工具坐标系
用户坐标系

ID 1
Group ID 1 ▾
名称

备注

* X -30554.203 mm

* Y 27925.616 mm

* Z 14674.516 mm

* A 19.572 °

* B -34.793 °

* C 65.524 °

方法 7p ▾

点位 1 *

点位 2 *

点位 3 *

点位 4 *

坐标原点 *

X方向 *

Z方向 *

点位 1 数据记录完毕

▶ 示教记录

✕ 取消

✓

✕

图 2.16 工具坐标系设置界面 4

点位记录完成后，“*”会由红色变成绿色

点位记录成功后会提示“点位 x 数据记录完毕”，此时可进行下一点位的示教

4. 根据图 2.12 点位要求依次示教 7 个点位。

用户坐标系

工具坐标系

ID 1	Group ID	1	名称	
备注				
* X	-4001.82	mm	* Y	1630.501 mm
* Z	1559.442	mm		
* A	-132.544	°	* B	22.967 °
* C	-175.024	°		

方法 7p ▾

点位 1 *
点位 2 *
点位 3 *
点位 4 *

坐标原点 *
X方向 *
Z方向 *

坐标系数据已经计算完毕，确认后请保存

示教记录
取消

✓
✕

图 2.17 工具坐标系设置界面 5

5.7 个点位示教完成后，会提示“坐标系数据已经计算完毕，确认后保存”，此时点击图 2.17 中的，会提示“坐标系保存成功”，如图 2.18 所示，此时工具坐标系已设置完成。

☰	admin ▾	k_main : 11	UF:0 ▾	组别:1	关节坐标系 ▾	60% 不限速	<div style="border: 1px solid green; padding: 2px; color: green; font-weight: bold; font-size: small;"> 成功 坐标系保存成功 </div>
	2023-02-07 17:26:49	Operation-0028	TF:0 ▾	ON_STANDBY	连续运行		

工具坐标系
用户坐标系

TF[1]	ID 1	Group ID	1	名称	
	备注				
	+ X	-4001.82	mm	+ Y	1630.501 mm
	+ Z	1559.442	mm		
	+ A	-132.544	°	+ B	22.967 °
	+ C	-175.024	°		

编辑

图 2.18 工具坐标系设置界面 6

六轴机器人四点法设置工具坐标系

四点法设置工具坐标系步骤：进入工具坐标系设置界面，选择四点法，根据点位要求示教图 2.12 中的 P1-P4 点即可，具体操作步骤请参考七点法

四轴机器人四点法设置工具坐标系步骤

1. 依次点击“菜单按钮”→“坐标系”→“工具坐标系”进入界面如图 2.19 所示，然后选择 4p 方法。

图 2.19 工具坐标系设定界面

2. 如图 2.20 所示，操作坐标系选择为“基座坐标系”，将工具沿四个不同的姿态方向移至同一个参考点如图 2.21 所示，分别记录点位。

☰	admin ▾ 11-05-2020 12:58:49	无程序运行 System-0084	UF:0 ▾ TF:0 ▾	组别:1 ON_STANDBY	底座坐标系 ▾ 连续运行	25% 不限速
---	--------------------------------	----------------------	------------------	--------------------	-----------------	------------

工具坐标系
用户坐标系

TF[1:default]

🗑️ 删除
➕ 新建

ID 1	* 名称	default			
Group ID	1 ▾	备注	default		
* X	0	mm	* Y	0	mm
* Z	0	mm	* A	0	°
* B	0	°	* C	0	°

方法 4p ▾

点位 1

已记录

点位 2

*

点位 3

*

点位 4

*

点位 1 数据记录完毕

🕒 示教记录
✕ 取消

✓ 保存

图 2.20 工具坐标系设定界面

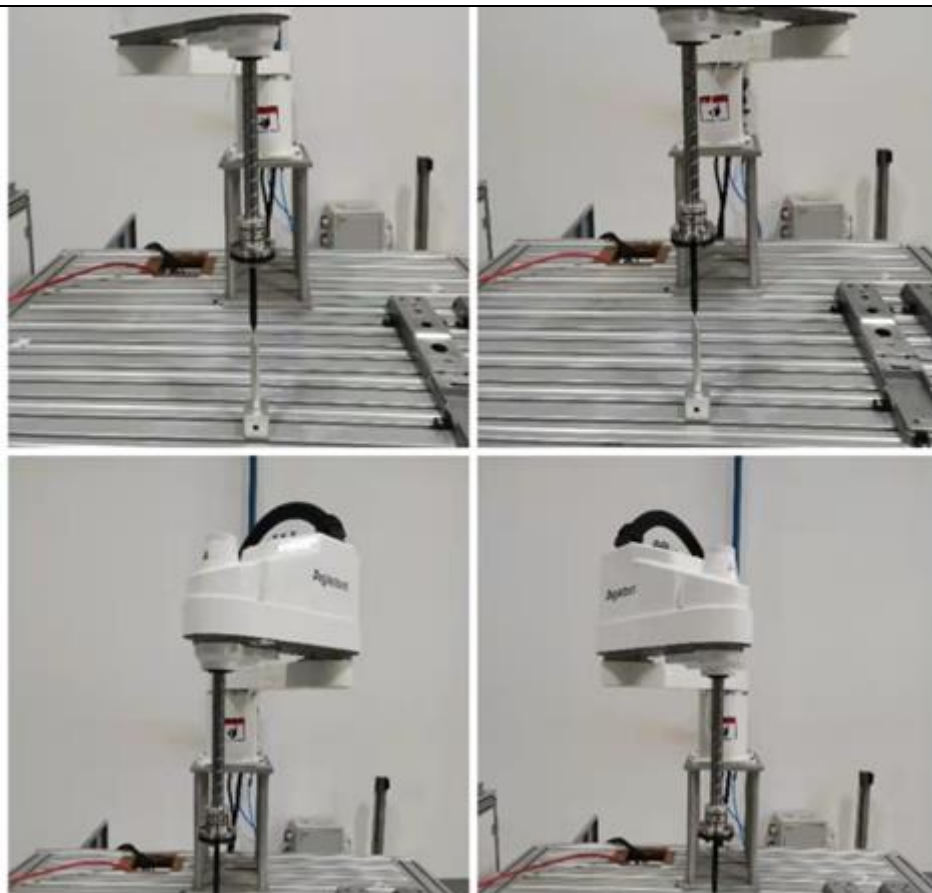


图 2.21 四点法参考位置

3. 点位记录结束，工具坐标系将被自动计算如图 2.22 所示，点击“保存”按钮即可记录坐标系设定。

☐	admin ▾	无程序运行	UF:0 ▾	组别:1	基座坐标系 ▾	25% 不限速
	11-05-2020 13:01:42	System-0084	TF:0 ▾	ON_STANDBY	连续运行	

工具坐标系 用户坐标系

TF[1:default]

删除
新建

ID 1	*名称	default			
Group ID	1 ▾	备注	default		
* X	0	mm	* Y	0	mm
* Z	0	mm	* A	0	°
* B	0	°	* C	0	°

方法 4p ▾

点位 1 已记录

点位 4 已记录

点位 2 已记录

点位 3 已记录

坐标系数据已经计算完毕，确认后请保存

示教记录
取消

保存

图 2.22 完成示教后点击保存

直接输入法设置工具坐标系

如图 2.23 所示，选择直接输入法，写入工具坐标系数据 X,Y,Z,A,B,C

☐	admin ▾	无程序运行	UF:0 ▾	组别:1	基座坐标系 ▾	25% 不限速
	11-05-2020 13:05:37	System-0084	TF:0 ▾	ON_STANDBY	连续运行	

工具坐标系 用户坐标系

TF[1:default]

删除
新建

ID 1	*名称	default			
Group ID	1 ▾	备注	default		
* X	0	mm	* Y	0	mm
* Z	0	mm	* A	0	°
* B	0	°	* C	0	°

方法 Direct ▾

保存

图 2.23 直接写入法窗口

2.2.2 设定用户坐标系

用户坐标系即为根据用户的作业空间定义的直角坐标系。

用户坐标系在尚未设定时，默认用户坐标系 UF[0]与基坐标系重合。

用户坐标系，通过相对基坐标系的坐标系原点的位置 (X,Y,Z) 和绕 X 轴、Y 轴、Z 轴的回转角 (A,B,C) 来定义。

用户坐标系，在设定和执行位姿寄存器，执行位置补偿指令时使用。另外，程序中的位置也可根据用户坐标进行示教。有关位置寄存器的设定，请参阅 5.1.3 节；有关位置补偿指令的执行，请参阅 3.3.5 动作附加指令。



以关节形式示教的情况下，即使变更用户坐标系也不会对位置变量、位置寄存器产生影响，但是以笛卡尔形式示教的情况下，位置变量、位置寄存器都会受到用户坐标系的影响，请予注意。

通过坐标系设定画面定义用户坐标系时，可定义 10 个坐标系，并可根据情况进行切换。

设定用户坐标系的优点：

- 可使工具坐标系沿着工作表面或工件的边缘手动移动。
- 可以以用户基坐标为参照进行示教点位。如果必须移动工件，例如由于工件工装夹具被移动，只需重新设定用户坐标系到工装夹具上面相同的位置，程序点位即可无需重新示教。
- 默认用户坐标系与基座坐标系重合

用户坐标系的数据格式为：[X 0mm,Y 0mm,Z 0mm,A 0°,B 0°,C 0°]

设定方法：

1. 三点法

如图 2.24 所示，第一个点“X Start”为 X 轴的起始点；第二个示教点“X Direction”为 X 轴方向的点；第三个示教点“Y Direction”为 Y 轴方向上的点；点击保存后系统会自动计算出该用户坐标系相关的 XYZABC 参数。

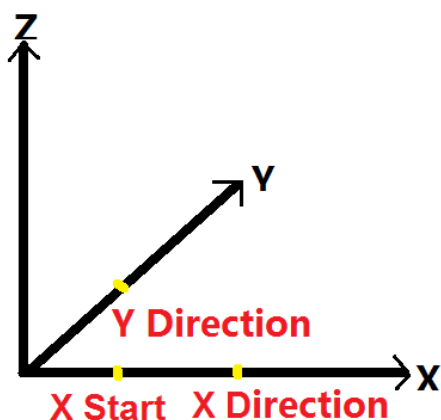


图 2.24 三点法

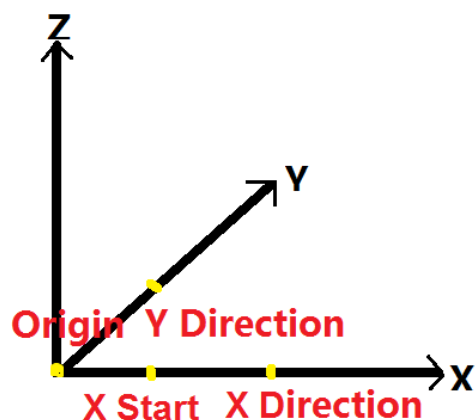


图 2.25 四点法

2. 四点法

如图 2.25 所示，第一个点“X Start”为 X 轴的起始点；第二个示教点“X Direction”为 X 轴方向的点；第三个示教点“Y Direction”为 Y 轴方向上的点；第四个点“Coord Origin”为该用户坐标系原点位置（可为空间中任意一点）；点击保存后系统会自动计算出该用户坐标系相关的 XYZABC 参数。

3. 直接输入法

三点法/四点法步骤：

1. 依次点击“菜单按钮”→“坐标系”→“用户坐标系”进入图 2.26 所示界面，选择 3p/4p 方法，如下图所示。

工具坐标系

UF[1:default]

删除
新建

ID 1 *名称 default

Group ID 1 备注 default

* X 0 mm * Y 0 mm * Z 0 mm

* A 0 ° * B 0 ° * C 0 °

方法 3p

Direct 点 *

3p 向 *

4p 向 *

Y方向 *

示教记录
取消

保存

图 2.26 用户坐标系设定窗口

2. 按三点法/四点法描述，记录点位，待所有记录点位结束后，即可自动计算用户坐标系，如下图 2.27 三点法/2.28 四点法所示。

☐☐☐☐	admin ▾	● 无程序运行	UF:0 ▾	组别:1	基座坐标系 ▾	25% 不限速
	11-05-2020 13:09:42	System-0084	TF:0 ▾	ON_STANDBY	连续运行	

工具坐标系

UF[1:default]

删除
新建

ID 1 *名称 default

Group ID 1 备注 default

* X 0 mm * Y 0 mm * Z 0 mm

* A 0 ° * B 0 ° * C 0 °

方法 3p

X始点
已记录

X方向
已记录

Y方向
已记录

坐标系数据已经计算完毕，确认后请保存

示教记录
取消

保存

图 2.27 三点法设定窗口

	admin	无程序运行	UF:0	组别:1	底座坐标系	25%不限速
11-05-2020 13:10:22	System-0084	TF:0	ON_STANDBY	连续运行		

用户坐标系 工具坐标系

UF[1:default]

ID 1	*名称	default						
Group ID	1	备注	default					
*X	321.315	mm	*Y	0	mm	*Z	870.097	mm
*A	0	°	*B	0	°	*C	0	°

方法 4p

X始点

已记录

坐标原点

已记录

X方向

已记录

Y方向

已记录

坐标系数据已经计算完毕，确认后请保存

图 2.28 四点法设定窗口

直接输入法:

在用户坐标系界面，选择直接输入法（Direct），写入用户坐标系数据。

	admin	无程序运行	UF:0	组别:1	底座坐标系	25%不限速
11-05-2020 13:16:43	System-0084	TF:0	ON_STANDBY	连续运行		

用户坐标系 工具坐标系

UF[1:default]

ID 1	*名称	default						
Group ID	1	备注	default					
*X	0	mm	*Y	0	mm	*Z	0	mm
*A	0	°	*B	0	°	*C	0	°

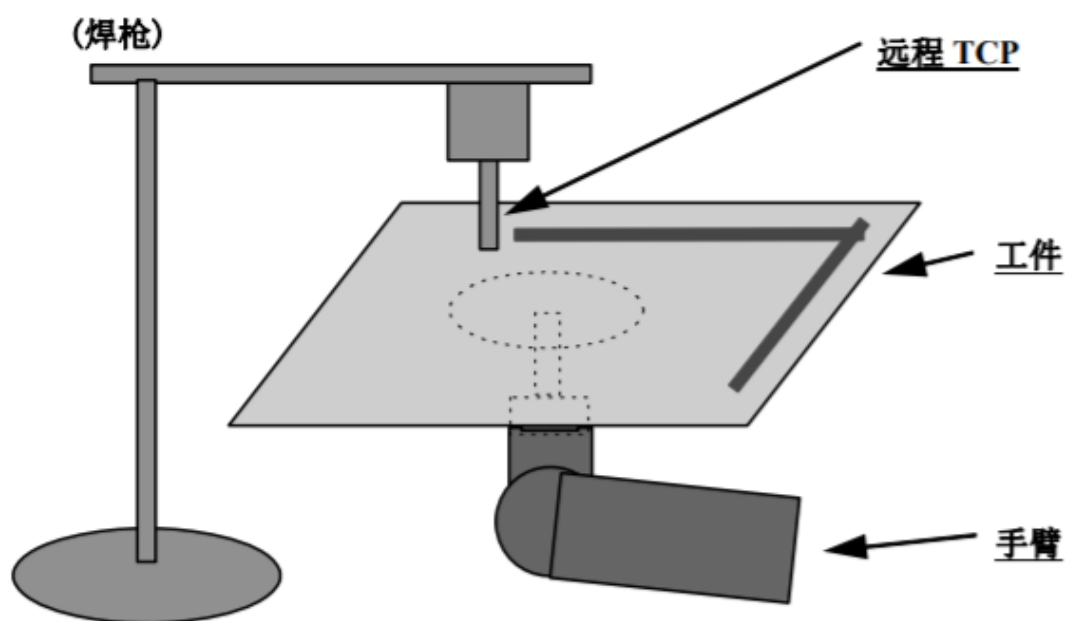
方法 Direct

图 2.29 直接输入法设定窗口

2.2.3 设定 Remote TCP

Remote TCP 概述：

使用远程 TCP 功能，则可相对固定在地面的 TCP（即远程 TCP），移动固定于机器人的工件，即相对于固定在地面的工具使工件执行直线、圆弧等动作，并由此进行加工。



远程 TCP 功能利用例(焊封应用)

图 2.30 远程 TCP 效果图

使用远程 TCP 功能，需要对机器人示教相对地面被固定的工具的突出点位置。

RTCP 坐标系的设定方法与用户坐标系的设定方法一致，均采用直接输入法/三点法/四点法，可直接在用户坐标系的配置页面中设定 RTCP 坐标系。

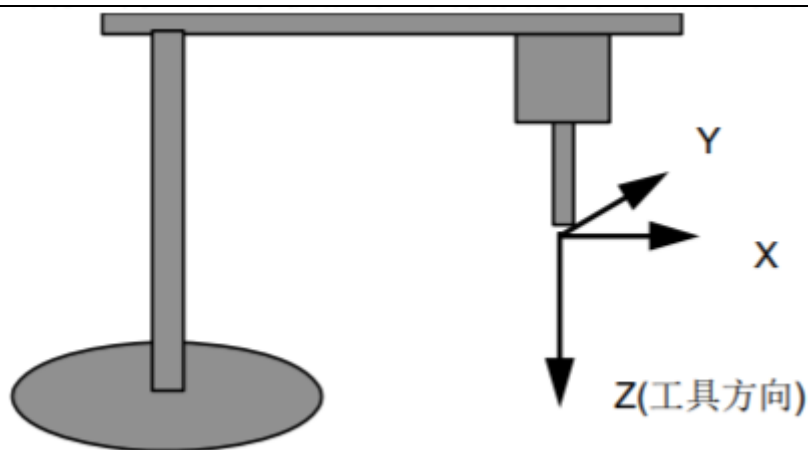


图 2.31 远程 TCP 坐标系示意图

当想要基于 RTCP 坐标系进行点动示教时，即围绕远程 TCP 进行平行移动/回转，则需在页面状态栏中坐标系一栏下拉选择“RTCP/工具”或“RTCP/用户”，并选择相应的用户坐标系及工具坐标系。



图 2.32 参考坐标系选择菜单

“RTCP/工具”及“RTCP/用户”的特点及区别：

坐标系	围绕点	平行移动/回转方向
RTCP/工具	RTCP	所选的工具坐标系
工具坐标系	TCP	所选的工具坐标系
RTCP/用户	RTCP	所选的用户坐标系

图 2.33 远程 TCP 移动方向参考表

在执行远程 TCP 动作时，需在指令中添加远程工具动作附加指令 RTCP。当运动指令使用 RTCP 附加指令时，记录点位时所使用的用户坐标系就会变成 RTCP 坐标

系，若不带 RTCP 附加指令，则依旧保持为用户坐标系。



RTCP 不能在关节动作下使用。

例子：

相对远程 TCP，在 2000mm/s 的相对速度下将工件移动到 P[1]点：

```
MoveL P[1] 2000mm/s Fine RTCP
```

相对远程 TCP，在 2000mm/s 的相对速度下使工件经由 P[1]移动到 P[2]：

```
MoveC P[1] P[2] 2000mm/s Fine RTCP
```

2.2.4 坐标系转换功能

坐标系转换概述：

坐标系变换功能，指的是选取程序中某一范围的指令，针对其中已经记录好位姿数据的运动语句，变换位姿数据中所使用的工具坐标系或用户坐标系，并将变换后的程序语句作为新的程序或程序段。在变换前，可通过选择不同的变换规则，使得位姿数据中的坐标值更改或保持不变，即可让机器人基于新坐标系移动到新的实际位置，也可让机器人的实际位置保持和变换前一致，即各关节角度保持不变。

应用场景：

原先使用的机械手 A 损坏了，需要用机械手 B 做替换，但机械手 B 比机械手 A 短 1cm，则此时可先建立机械手 B 的工具坐标系，然后配置工具坐标系变换规则并执行变换，最后当机器人执行变换后的程序时，其使用机械手 B 也能到达原先使用机械手 A 所能到达的位置。

当需复制工作面及其轨迹时，可先基于新工位建立用户坐标系，然后配置用户坐标系变换规则并执行变换，最后当机器人执行变换后的程序时，其在新用户坐标系下的相对轨迹可保持与其在原工作面中的轨迹一致。

坐标系变换的设置：

1. 依次点击“菜单按钮”→“应用”→“坐标系变换”进入坐标系变换界面，如图 2.34 所示。

坐标系变换

原始程序	<input type="text" value="请选择"/>	目标程序	<input type="text"/>
变换范围	<input type="text" value="局部"/>	插入行	<input type="text" value="1"/>
程序范围	<input type="text" value="1 至 -1"/>		

变换目标	<input type="text" value="工具坐标系"/>	变换规则	<input type="text" value="TCP固定"/>
变换前工具坐标系编号	<input type="text" value="请选择"/>	变换后工具坐标系编号	<input type="text" value="请选择"/>

图 2.34 坐标系变换界面

坐标系变换参数说明：

- 原始程序：需要进行坐标系变化的程序
 - 目标程序：在此处为变换后的新程序命名也可以使用原始程序名
 - 变换范围：可以选择全部，把所有整个程序进行变换，还可以选择局部，变换指定的程序范围行
 - 程序范围：与变换范围配合使用
 - 插入行：插入目标程序的第几行
 - 变换目标：有用户坐标系与工具坐标系两种
 - 变换规则：若变换目标为用户坐标系，则变换规则为不变换位姿数据与变换位姿数据两种
 - 若变换目标为工具坐标系，则变换规则为 TCP 固定与机器人固定两种
 - 变换规则具体见下文中的变换规则详细说明
 - 变换前工具/用户坐标系编号：原程序中的点位使用的工具/用户坐标系编号
 - 变换后工具/用户坐标系编号：目标程序将要使用的工具/用户坐标系编号
2. 以上参数设置完成后，点击“执行变换”出现下面弹窗



图 2.35 变换完成指示窗

3. 点击“保存”即可完成变换操作

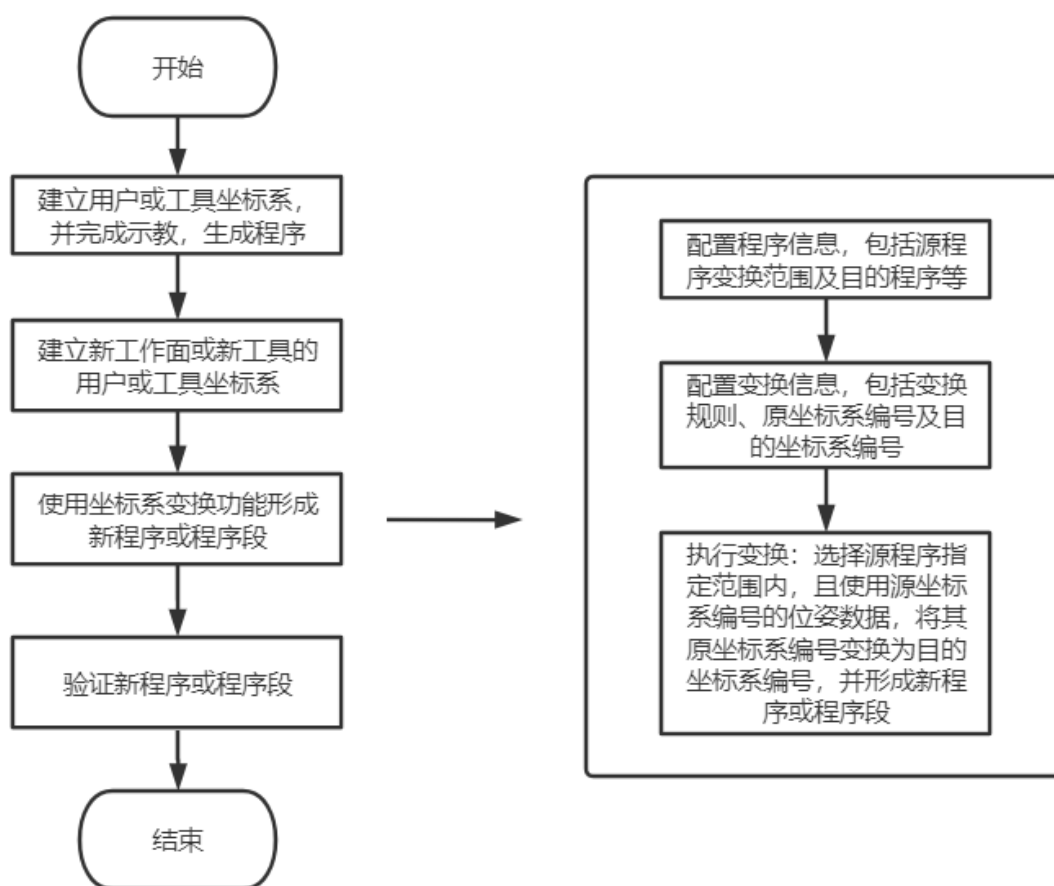


图 2.36 坐标系变化流程图



位姿数据中的位置和姿态数值:

仅针对程序中的 P[] 进行偏移计算, 位姿寄存器 PR[] 保持不变, 不参与偏移计算。

经过偏移计算后，基于直角坐标系表达的位姿 $P[]$ 还是直角坐标值，基于关节坐标系表达的位姿 $P[]$ 还是关节坐标值。

当经过偏移计算后的 $P[]$ 落在运动空间之外时，若该 $P[]$ 基于关节坐标系表达，则作为一个未示教记录的值进行保存；若该 $P[]$ 基于直角坐标系表达，则直接保存偏移后的数值。

工具坐标系变换中的变换规则：

- TCP 固定：变换前后，位姿数据中，“变换后工具坐标系编号”直接替代“变换前工具坐标系编号”，且位姿坐标值保持不变。典型应用场景：用于更换损坏的工具，使用该规则，可保证使用新工具执行变换后的程序时仍能到达原工具的工作位置。
- ROBOT 固定：变换前后，位姿数据中，“变换后工具坐标系编号”直接替代“变换前工具坐标系编号”，但位姿坐标值会被重新计算并生成，其计算规则是机器人相对原有位姿的关节角度保持不变。典型应用场景：注重让机器人避让某些路径位姿，但对 TCP 所处位置不在意的场合

用户坐标系变换中的变换规则：

- 不变换位姿数据：变换前后，位姿数据中，“变换后用户坐标系编号”直接替代“变换前用户坐标系编号”，且位姿坐标值保持不变。典型应用场景：原有工作面发生位移或需复制一个工作面时，可先基于变化后的工作面建立新的用户坐标系（已基于原工作面建立用户坐标系），然后使用该规则并执行用户坐标系变换，最后当机器人在执行变换后的程序时，可保证机器人相对新工作面的路径不变。
- 变换位姿数据：变换前后，位姿数据中，“变换后用户坐标系编号”直接替代“变换前用户坐标系编号”，但位姿坐标值会被重新计算并生成，其计算规则是保持机器人相对原有位姿的关节角度不变。典型应用场景：目标工件的实际位置未发生改变，只是工件所依赖的基准发生了改变。

2.3 软限位

通过软件对机器人各轴关节的运动范围进行限制保护，使其默认软件保护范围小于机型默认的硬限位，避免正常使用中经常发生硬限位的撞击。

允许用户根据情况对软限位的关节范围进行设置（但始终小于机型的机械硬限位范围）。



1 请勿仅仅依靠关节可动范围功能来控制机器人的动作范围。应同时使用极限开关和硬限位。否则，恐会导致人员受伤，或损坏装置。

2 请对硬限位进行调节，以使其适合软件的调节。否则，恐会导致人员受伤，或损坏装置。



关节可动范围的变更，对机器人的动作范围产生影响。为避免故障，在变更各轴可动范围之前，需要重新考虑其造成的影响。若不加充分考虑变更可动范围，则有可能导致在以前示教的位置发生报警等意想不到的结果。

上限值：

表示关节可动范围的上限值。这是正方向的可动范围。

下限值：

表示关节可动范围的下限值。这是负方向的可动范围。

功能描述：

软限位的关节范围值可以通过 TP 应用端进行设置。设置方式为输入框修改，精度为 0.001° 。禁止空值、字符。禁止填写的软限位范围值超出出厂默认值范围（默认范围显示在软限位设置界面中）。

2.3.1 软限位界面

设置关节软限位步骤：

依次点击“菜单按钮”→“系统”→“基础设置”→“软限位设置”进入界面如图 2.37 所示。

组号 1 组名 GBT-S6A-600

轴	默认下限	软限位下限	软限位上限	默认上限
轴1	-132 °	<input type="text" value="-130"/> °	<input type="text" value="130"/> °	132 °
轴2	-150 °	<input type="text" value="-150"/> °	<input type="text" value="150"/> °	150 °
轴3	-200 mm	<input type="text" value="-200"/> mm	<input type="text" value="0"/> mm	0 mm
轴4	-360 °	<input type="text" value="-360"/> °	<input type="text" value="360"/> °	360 °

[编辑](#)

图 2.37 软限位设置界面

通过该界面，用户可以自行对机器人软限位进行设定，但用户设定的软限位需小于等于出厂时设定的默认软限位。

- 轴：显示该运动组下所有的轴信息。
- 软限位下限：轴限位下限，并不一定是负数，但必须小于等于正限位。
- 软限位上限：轴限位上限，并不一定是正数，但必须大于等于负限位。



点击“编辑”可修改软限位上下限；在正负限位输入框中，若用户填入大于出厂时规定的正限位，系统会自动更改为出厂时的正限位上限；若用户填入小于出厂时规定的负限位，系统会自动更改为出厂时的负限位下限。更改完成后点击保存即可立即生效。

2.4 负载设置

负载设定概要：

负载设定，安装在机器人上负载信息（重量、重心位置等）的相关设定。

通过适当设定负载信息，就会带来如下效果。

- 动作性能提高。(振动减小、循环时间改善、提高精度等。)
- 更加有效地发挥与动力学相关功能。(碰撞检测功能、重力补偿功能等的性能提高。)

如果负载信息错误变大，则有可能导致振动加大，或错误检测出碰撞。为了更加有效利用机器人，建议用户对配备在机械手、工件、机器人手臂上的设备等负载信息进行适当设定。

负载信息的设定，在“负载设置画面”上进行。使用该画面，可以设定 10 种负载信息。通过预先设定多个负载信息，只要切换负载设定编号就能对应负载的变更。此外，可通过程序指令，在程序中切换负载设定编号（见 3.8.5 节）。

机器人能够针对不同的负载选择或者调整内部的控制参数，使机器人能够达到相应负载下最优的控制精度和稳定性。

负载的设定步骤如下：

1. 依次点击“菜单按钮”→“系统设置”→“基础设置”→“负载设定”进入图 2.38 所示界面。默认激活负载设定为[Payload: 0],且不可编辑。

* ID	名称
0	Default

重量

M	2	kg
---	---	----

质心

x	65.906	mm
y	0	mm
z	-57.816	mm

转动惯量

Ix	0.008832	kg.m ²
Iy	0.02097	kg.m ²
Iz	0.012491	kg.m ²

★ 激活

图 2.38 负载参数界面

2. 点击“新建”新建负载设定后。点击“编辑”，即可手动输入负载数据，如图 2.39 所示。

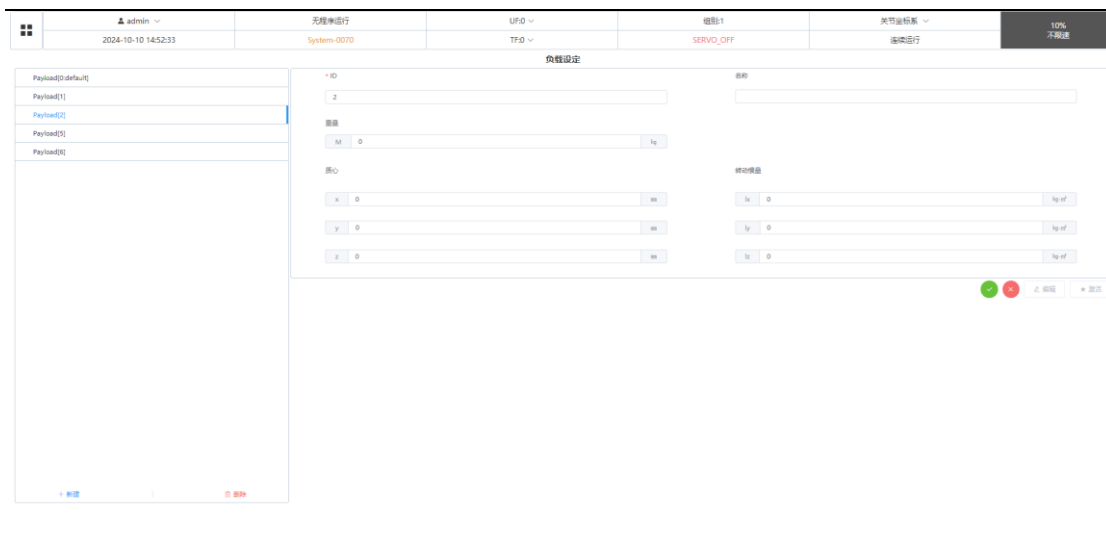


图 2.39 负载参数界面

其中 M (kg) 为负载的质量, X (mm)、 Y (mm)、 Z (mm)即如图 2.39 中 X 、 Y 、 Z 所示为负载相对于法兰中心的重心位置, I_{xx} 、 I_{yy} 、 I_{zz} 分别为负载相对于 X 、 Y 、 Z 坐标系方向的转动惯量; 当把机器人负载当作质点处理时, 转动惯量 I_{xx} 、 I_{yy} 、 I_{zz} 写成 0

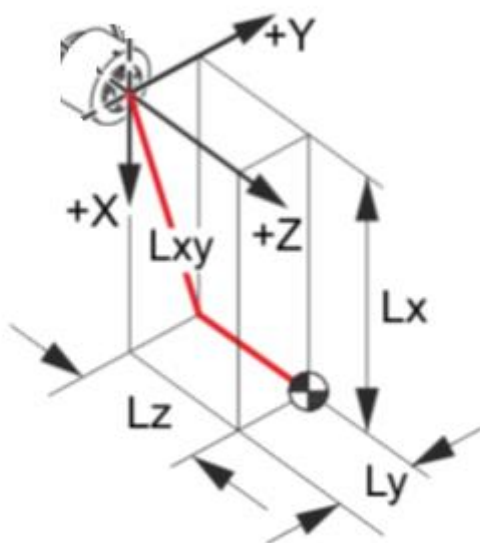


图 2.40 负载位置参考

3. 激活负载设定如图 2.40 所示, 激活的负载前有“√”标识。

2.5 零点标定

校准信息零点校准是指把每个机器人关节的角度与脉冲计数值关联起来的一种操作。

零点校准操作目的是获得对应于零点位置的脉冲计数值。

“零点校准”是在出厂前完成的。日常操作中没有必要执行零位校准操作。但是在下述情况下需要执行零点校准操作。

如遇以下情况：电机更换，脉冲编码器更换，减速器更换，机械本体中用于脉冲计数备份的电池电量用完，请联系我公司进行高精度标定操作。



包含零点标定数据在内的机器人的数据和脉冲编码器的数据，通过各自的备用电池进行保存。电池用尽时将会导致数据丢失。应定期更换控制柜和机构部的电池。电池电压下降时，系统会发出报警通知用户，请及时更换电池。

零点标定方法

- 一般标定法
- 零点编码数据直接写入法

2.5.1 一般标定法

选择某个或某几个轴，将它们当前的读数作为新的零点数据记录到机器人 Flash 的参数文件中。记录对象包括机器人本体轴和附加轴（如有）。可以标定单轴。（例如，用户移动机器人让某轴机械零刻度线重合，再使用此功能即可实现机器人的零点标定。）

由于用来后背脉冲计数器的电池电压下降，或更换脉冲编码器等而导致某一特定轴的零点标定数据丢失时，进行零点标定。标定方式选择一般标定法，勾选多个轴进行标定也可勾选单个轴进行标定。勾选完成点击标定按钮既完成标定。

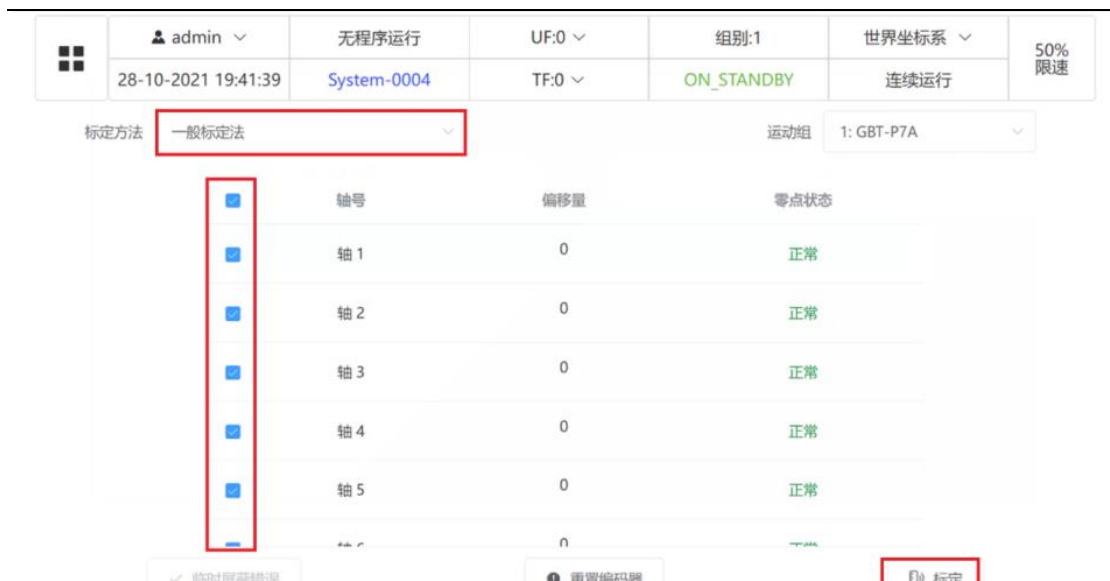


图 2.41 一般标定法界面

一般标定法的设定步骤如下：

1. 六轴机器人将各轴零点参考标记对齐，四轴机器人将一、二、四轴零点参考标记对齐，三轴升到最高处。
2. 依次点击“菜单按钮”→“系统”→“基础设置”→“零点设置”进入界面如图 2.42 所示。

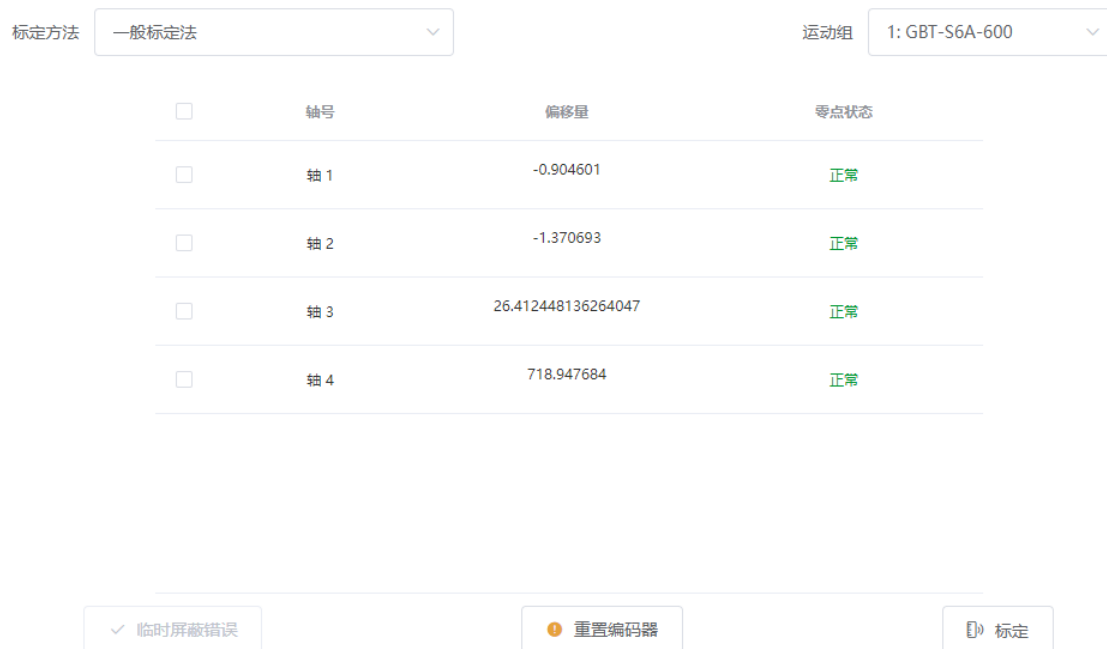


图 2.42 零点状态界面

3. 在零点标定界面左上角点击标定方法选择“一般标定法”。

标定方法 一般标定法

4. 通过点击轴号前的框，即可选择所有要标定的轴 2.43 所示；或者可以单独选择要标定的轴如图 2.44 所示。



图 2.43 选择多轴界面

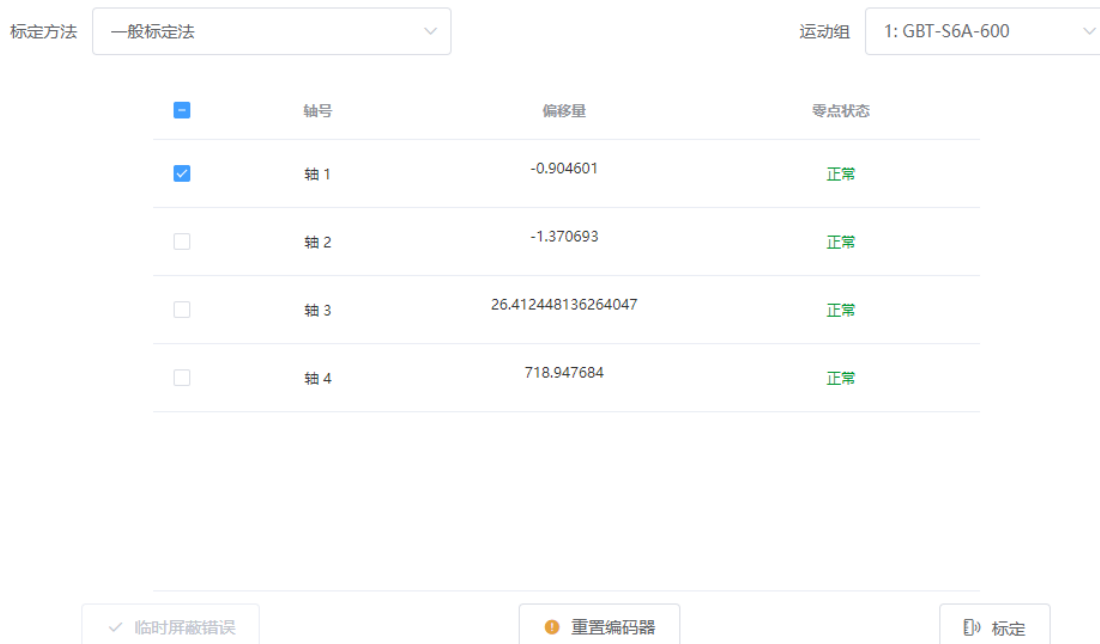


图 2.44 选择单轴界面

5. 需要标定的轴选择完成后点击“标定按钮”，如果成功弹出“标定成功”消息，对应标定的轴号零点状态变为“未保存”。如下图 2.45 所示。



图 2.45 状态变为为保存

6. 点击保存即可完成标定。

2.5.2 零点编码数据直接写入法

零点标定数据的直接输入，可将零点标定数据值直接输入到系统变量中。这一操作作用于零点标定数据丢失而脉冲数据仍然保持的情形。

零点编码数据直接写入法步骤如下：

1. 标定方法选择“零点编码数据直接写入法”，点击轴号前的选择框，选择需要标定的轴。
2. 偏移量内写入零点数据（注：六轴机器人偏移量对应的零点数据标注在机器人本体底座的标签上；四轴机器人没有）；然后点击“标定”，最后点击“保存”。

0

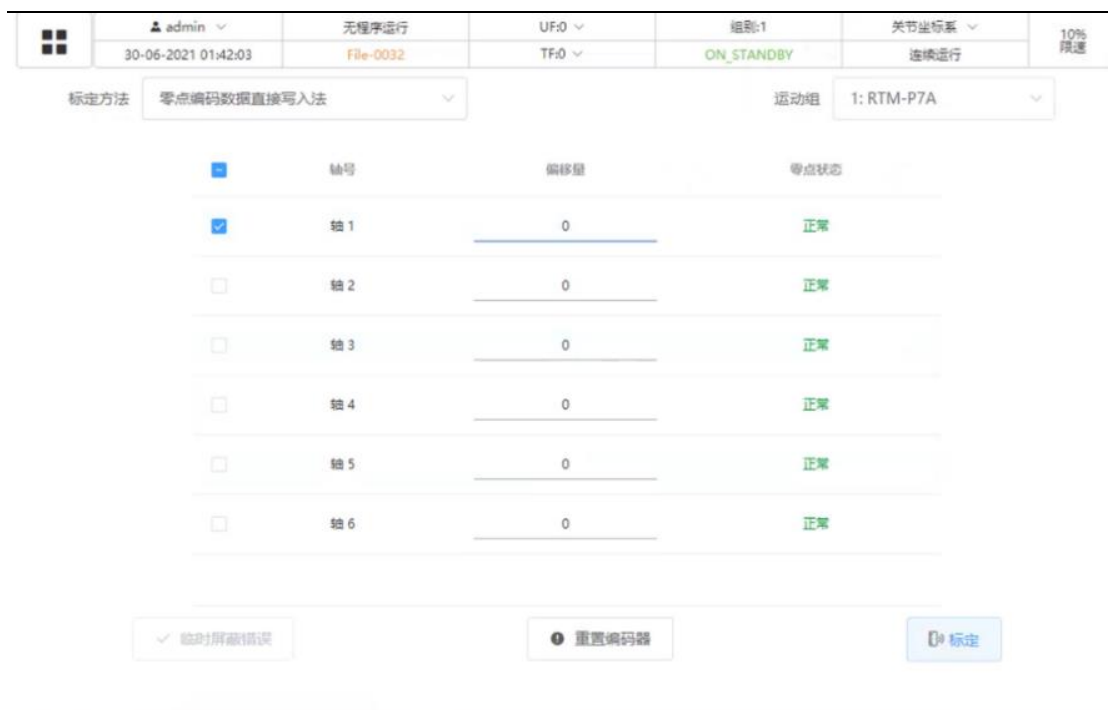


图 2.46 直接写入法窗口

2.5.3 临时屏蔽错误功能键

对于发现有零点错误的轴存在时，按下“临时屏蔽错误”键后（无需选中具体轴），发生报错的轴被屏蔽。使得机器人报警轴可以跳出报警状态再次运动（仅限关节示教运动）。

临时屏蔽错误功能键只对“丢失零点”标志的轴生效，其他轴继续沿用原先的“零点数据”。

一个问题轴重新做好零点标定并保存标定数据后，该轴的“临时状态”标志消失。当全部轴的零点丢失状态都消失后，即显示为“正常”标志，机器人就可以恢复正常运动模式了。

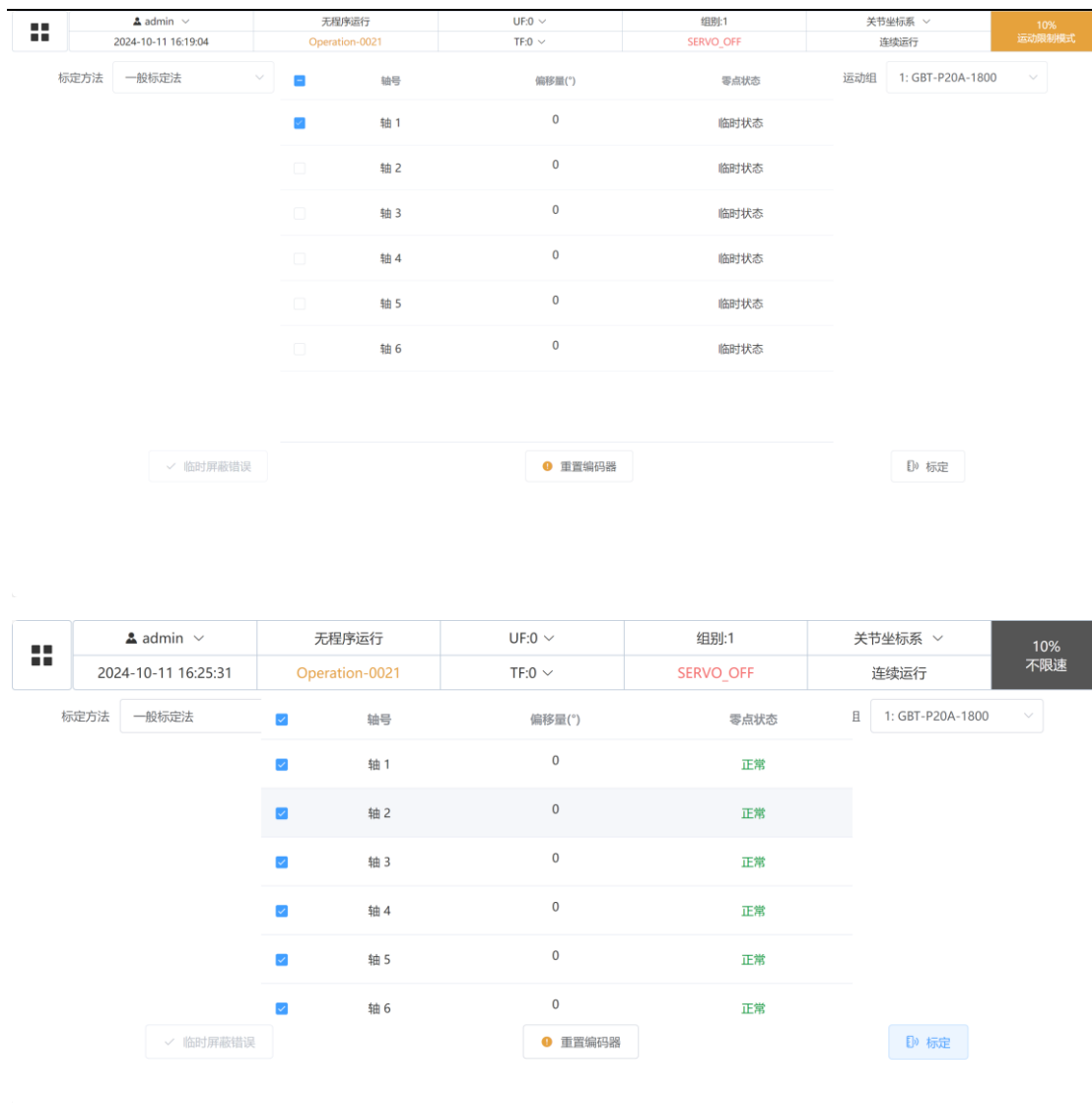


图 2.47 零点状态窗口

2.5.4 重置编码器/清编码器电池错误功能键

在编码器掉电或者重新拆装机器人本体后才需要清理编码器多圈信息，此时一般会出现“编码器电池错误”报警，注意仅有零点丢失报警的情况下使用一般标定法或者直接输入法操作即可。

按下“重置编码器”功能键（如图 2.47）之前需要注意必须在机器人所有轴都保持机械零位状态下进行这个操作（因为此功能键是面向机器人的所有轴）。如不按此操作，则可能导致之后的标定工作无法达到最佳准确度性能。



图 2.48 零点状态窗口

按下该按键后弹出警告对话框如图 2.48, “此操作将导致机器人所有轴零点丢失, 且之前标定的偏移值数据失效, 需要联系厂商专业标定才能恢复高准确度。请在符合《用户操作手册》或《报警手册》指导的情况下使用。请确认是否进行此操作?”

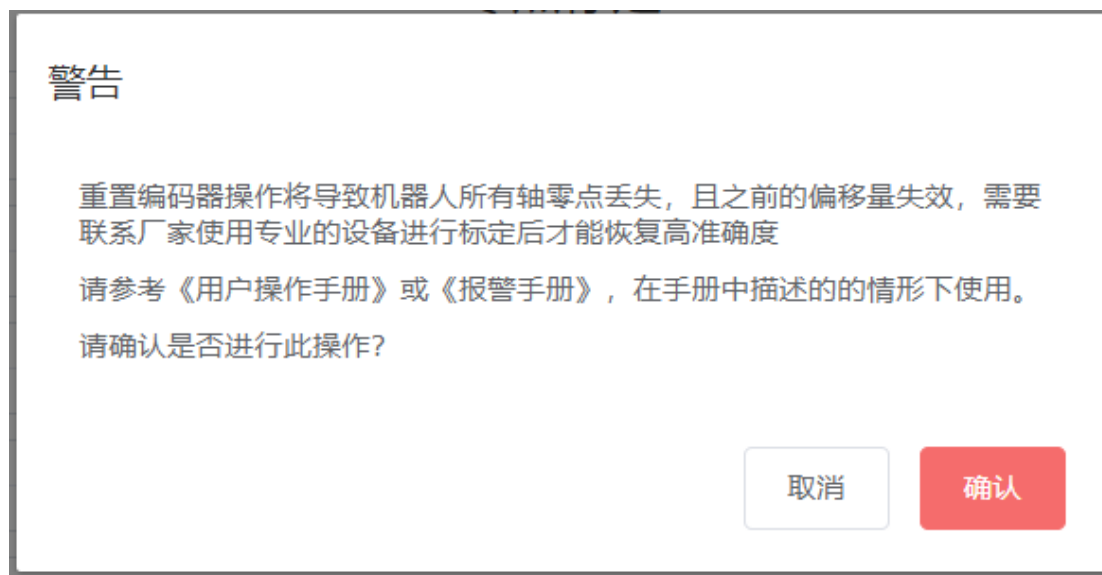


图 2.49 重置警告窗口

确认操作后, 清除所有编码器的多圈数据, 且无论之前零点丢失与否, 都将零点丢失标志置位, 要求用户进行零点标定。

不合理的清理多圈操作对机器人精度会产生影响, 需要慎重进行。如果用户不慎进行了清理多圈值操作, 将导致机器人准确度下降, 需用专用的标定仪器重新标定后方可恢复机器人的准确度。如有需要建议联系机器人厂家进行标定工作。

2.5.5 零点标定场景说明

机器人需要按下清理多圈值功能键的场景：

- 编码器电池耗尽（一般可以在报警栏上看到有电池欠压报警，类似：“Servo-4108 轴 1 编码器电池错误等”相似类型的报警）。此时用户更换电池后，需要先清理一下编码器多圈信息，然后再重新进行零点标定。
- 换电机时，需要拔掉电池线，所以编码器也会掉电；所以换电机后，在零点标定前，也需要先清理一下编码器多圈。

机器人不需要按下清理多圈值功能键的场景：

- 有可能出现零点丢失报警或零点异常警告，此种情况下，如遇到的是零点丢失报警则做零点标定即可；如遇到的是零点异常警告，则进入零点标定时将遇到“请人工确认零点是否正常的”询问弹窗，点击正常选项，警告自动消失系统恢复正常。

依次点击“菜单按钮”→“管理”→“时间/语言”进入如图 2.49 所示界面

用户可以更改当前的时间和切换中、英、韩、日、俄、越文语言。

	admin ▾	无程序运行	UF:0 ▾	组别:1	关节坐标系 ▾	20% 限速
	09-05-2020 14:56:22	Operation-0021	TF:0 ▾	ON_STANDBY	连续运行	

时间/语言设置

注意：时间设置成功后,必须关机并重启系统！

时间设置：

📅 2020年05月09日 ⌚ 14时52分59秒

语言切换：

图 2.50 时间语言设定窗口



时间的设置都必须关机并重启系统才能完成设置。

2.6 一般设定

2.6.1 时间/语言设置

依次点击“菜单按钮”→“管理”→“时间/语言”进入如图 2.49 所示界面 用户可以更改当前的时间和切换中、英、韩、日、越、俄文语言。



图 2.49 时间语言设定窗口

2.6.2 屏幕亮度设置

依次点击“菜单按钮”→“系统”→“其他设置”→“屏幕亮度设置”进入如图 2.50 所示界面

用户可以选择屏幕的亮度（默认从 1-255）

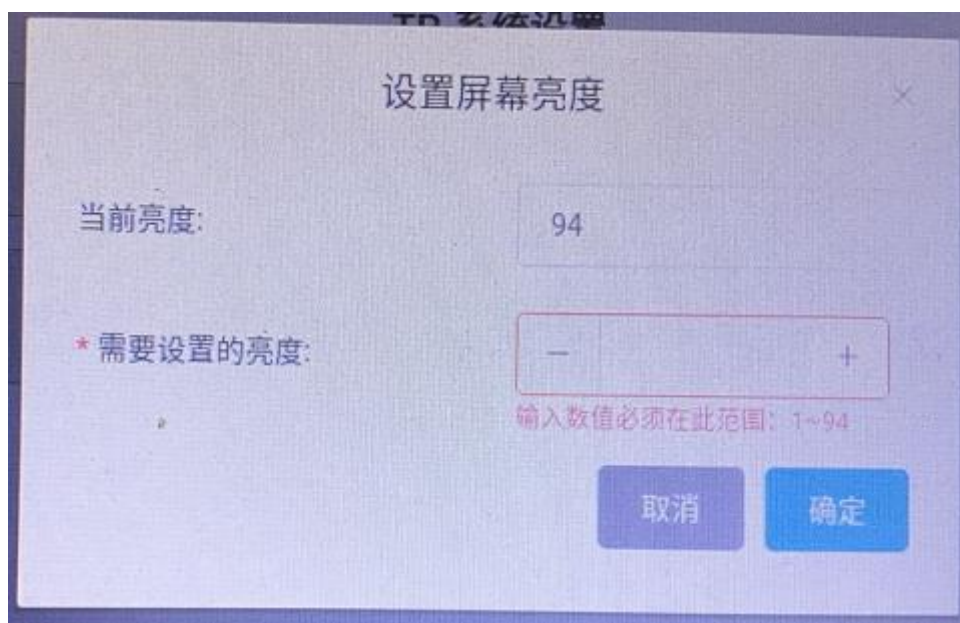


图 2.51 示教器显示屏上屏幕亮度设定窗口

2.6.3 屏幕自动熄屏

如果用户无操作超过一定时间（默认 10 分钟），屏幕会自动熄屏。

熄屏后，用户可以点击“锁屏”按钮解锁屏幕，幕将会重新点亮。

用户也可以在“菜单按钮->系统->其他设置->使用偏好”中修改自动熄屏时间。

2.6.4 IP 地址修改

当机器人独立部署，没有联网到现场的子网中时，直接使用默认的 IP，即可工作。

当需要把机器人融入现场的某个子网时，可能需要根据子网的 IP 范围，修改控制器和 TP 的 IP。

当换了新 TP 时，第一次启动，可能因为 TP 配对的控制器 IP 配置不正确，导致启动自检失败。仅限更新非默认 IP 地址（192.168.110.102）的 TP 或非默认 IP 地址（192.168.110.2）的控制柜的情形。此时界面上会有选择控制器的按钮，用户可以点击，来重新配置 TP 要连接的控制器 IP 地址。

如遇到问题，请联系厂商。

IP 地址修改步骤



点击`修改 TP IP`



修改 TP IP 地址为子网的某个空闲 IP
 依次点击“菜单按钮”→“系统”→“其他设置”→“修改 TP IP”进入修改 TP IP 界面



点击`配置控制器`



点击刷新列表，等待查找网络中的控制器



确认查找到的控制器 IP 为当前 TP 所连接的控制器，点击`连接`



点击`确认`即可，随后按提示操作`重启整个系统`。



点击`确认`即可，随后按提示操作`重启整个系统`。



如果修改地址为不同的子网。建议先修改控制器的 IP。再修改 TP 的 IP。否则处于不同的子网，有可能出现连接不上的情况。

IP 修改完成后要重启机器人控制柜与示教器。

2.6.5 查找控制器

此项功能用来查找与示教器相同网段下有哪些控制器，查找到控制器后还可以连接控制器。该功能多用于一个示教器适配多个控制柜情况下。

依次点击“菜单按钮”→“系统”→“其他设置”→“查找控制器”进入查找控制器界面，如图 2.52 所示。



图 2.52 查找控制器界面

在图 2.51 点击左上角“刷新列表”可刷新当前网络中存在的控制器数量；点击“修改”按钮可修改控制器的 IP 地址，点击“连接”按钮可使示教器连接到对应控制器。

2.6.6 选择控制器

此项功能用来选择与示教器相同网段下的控制器，该功能多用于一个示教器适配多个控制柜情况下。

依次点击“菜单按钮”→“系统”→“其他设置”→“选择控制器”进入选择控制器界面，如图 2.53 所示。在“新选择的控制器”中输入想要选择控制器的 IP 地址后点击“确定”即可。

图 2.53 选择控制器界面

2.7 系统参数

2.7.1 类型

- 机型参数：仅和产品型号相关，源自于设计开发所生成的设计参数。它可能是机械参数、控制柜参数、校核的性能参数。同一机型的机型参数，其格式不变、数值不变，除非机型本身发生设计变化。
- 本机参数：仅和具体的某一产品相关的物理参数，或者说无法由设计理论给出准确数值的某些参数。它是由实际生产制造的不一致性所造成的，不可避免，每台机器人的这类参数的值都是不一样的。
- 用户参数：提供外部接口，允许用户经常更改数值（用户指终端用户或者功能开发人员，并非所有参数都对终端用户开放），面向应用使用的参数。根据应用需要。

- 临时参数：为了满足系统中某些机制或软件功能而临时记录的参数数据，该参数如果需要掉电保存一般被记录在 NV-RAM 中，否则就在内存中。它只和某个时刻机器人的状态有关，与机器人本身的物理属性、算法变量、用户设置无关。

2.7.2 一般系统变量设置

依次点击“菜单按钮”→“系统”→“一般系统变量”进入如下图 2.54 所示界面。



图 2.54 系统变量界面

一般系统变量参数说明：

一般系统变量内容	意义
TP 手动模式指示信号	选择配置完成的数字输出信号，当示教器模式开关在手动全速模式下时，该信号状态为 ON，否则为 OFF
TP 手动限速模式指示信号	选择配置完成的数字输出信号，当示教器模式开关在手动限速模式下时，该信号状态为 ON，否则为 OFF
自动模式指示信号	选择配置完成的数字输出信号，当示教器模式开关在自动模式下时，该信号状态为 ON，否则为 OFF
TP 脱机模式指示信号	控制柜未插示教器

位姿/角度 Fine 到位检查稳定时间	位姿/角度 Fine 到位检查稳定时间，单位 ms
当前位姿寄存器传输有关节传输与笛卡尔传输两种类型	当为关节传输时，所配置 PR 寄存器中的数据的是机器人当前关节数据，当为笛卡尔传输时，所配置 PR 寄存器中的数据的是机器人当前位姿数据，该 PR 寄存器含位姿参数、回转数、关节配置在内的全部 PR 参数。该 PR 寄存器中的数据将每 8ms 更新一次
当前 TCP 末端速度	配置哪个寄存器，该寄存器将表示当前 TCP 速度，每 8ms 更新一次
视觉控制器 IP	连接视觉控制器的默认地址，修改后重启生效。
抖动抑制	抖动抑制功能开关。

2.8 用户级别

某些功能和设置只能由具有适当级别的用户访问。系统内有三个用户级别：

- 操作员
- 工程师
- 管理员

级别默认密码：

除操作员外，所有级别都需要密码才能激活。不同的用户有不同的操作权限。（不同用户的操作权限见安全使用须知中作业人员系统权限部分内容）

操作员没有默认密码，其他用户的默认密码如下表所示所示。

用户类型	默认密码
调试工程师 (engineer)	(可在 admin 权限下修改该密码)
管理员 (admin)	123

登录用户步骤如下：

1. 点击示教器状态栏中的“登录”进入登陆界面，如下图所示

登 录 ×

* 用户名

* 用户密码

[忘记密码?](#)

图 2.55 登录界面

2. 点击“请选择”选择需要登陆的用户类型
3. 输入对应类型的用户密码，点击“确认”即可完成登录

调试工程师修改密码步骤

1. 登录 admin
2. 依次点击“菜单按钮”→“用户管理”进入用户编辑界面，如图 2.56 所示

	admin <input type="text"/>	无程序运行	UF:1 <input type="text"/>	组别:1	关节坐标系 <input type="text"/>	10% 限速
	2023-03-27 14:48:44	Operation-0021	TF:1 <input type="text"/>	SERVO_OFF	连续运行	
						<input type="button" value="增加用户"/>

用户名	角色	说明	操作
1 admin	administrator	拥有超级权限	<input type="button" value="编辑"/> <input type="button" value="删除"/>
2 engineer	engineer	调试工程师	<input type="button" value="编辑"/> <input type="button" value="删除"/>

图 2.55 用户操作界面

3. 点击 engineer 后的“编辑”进入编辑用户界面，如图 2.57 所示

编辑用户 ×

* 用户名

* 用户权限

说明

* 密码

修改密码

图 2.57 编辑用户界面

4. 勾选修改密码，进入修改密码界面，如图 2.58 所示

The screenshot shows a web form titled "编辑用户" (Edit User) with a close button (X) in the top right corner. The form contains the following fields and options:

- * 用户名: engineer
- * 用户权限: engineer (dropdown menu)
- 说明: 调试工程师
- * 密码: 请输入当前登录用户的密码
- 修改密码
- * 新密码: 请输入新密码
- * 确认新密码: 请重复输入新密码

At the bottom right, there are two buttons: "× 取消" (Cancel) and "✓ 确认" (Confirm).

图 2.58 修改密码界面

5. 在“密码”处输入用户的登录密码，系统初始默认密码为“123”，然后输入“新密码”与“确认新密码”后，点击“确认”即可完成密码的修改

2.9 维护保养

菜单→系统→保养维护，进入设备维护保养信息界面。

保养信息

保养记录：记录了执行保养时间及下次保养时间。

设备记录：显示出厂时间、本次开机时间等。

← 返回首页
维护保养
刷新

保养信息
提示日志

保养记录

执行保养时间	2000-01-01
下次保养时间	2001-01-01

设备记录

出厂时间	2000-01-01
本次开机时间	2025-08-12 09:57:41
累计运行时间	2h

提示日志

← 返回首页
维护保养
刷新

保养信息
提示日志

序号	时间	提示信息
暂无数据		

< > 前往 1 页

3. 程序的构成

本章就程序的构成和程序指令进行说明。

机器人应用程序，由机器人为进行作业而由用户记述的指令、以及其它附带信息构成。

程序除了记述机器人如何进行作业的程序信息外，还有就程序属性进行定义的程序详细信息。



程序属性

程序适配机型: GBT-P7A-700

程序名称: k_main

程序备注: 程序备注支持不超过1000个字节的内容。

程序类型: None

创建时间: admin 创建于 2023-02-07 09:21:34

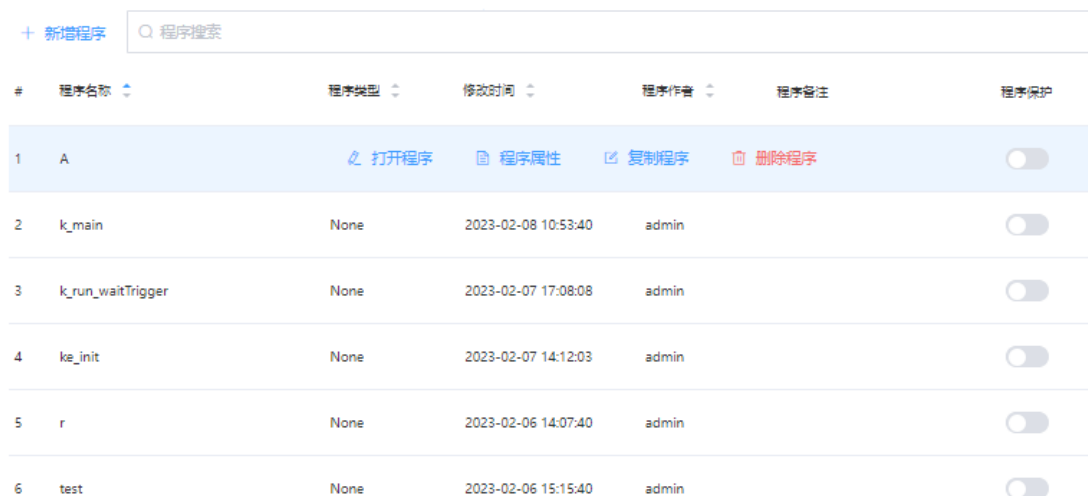
修改时间: admin 修改于 2023-02-07 17:17:39

X 取消 ✓ 确认

图 3.1 程序属性界面

程序属性由如下信息构成：

程序名称、程序备注、程序类型、创建时间、修改时间



#	程序名称	程序类型	修改时间	程序作者	程序备注	程序保护
1	A					<input type="checkbox"/>
2	k_main	None	2023-02-08 10:53:40	admin		<input type="checkbox"/>
3	k_run_waitTrigger	None	2023-02-07 17:08:08	admin		<input type="checkbox"/>
4	ke_init	None	2023-02-07 14:12:03	admin		<input type="checkbox"/>
5	r	None	2023-02-06 14:07:40	admin		<input type="checkbox"/>
6	test	None	2023-02-06 15:15:40	admin		<input type="checkbox"/>

图 3.2 程序列表界面



图 3.3 程序编辑语句界面

程序由如下信息构成：

- 赋予各程序指令的行号码。
- 机器人指令向哪个方向如何运动的动作指令。
- 包括如下内容的程序指令。
 - 存储数值数据的寄存器指令。
 - 存储机器人位置资料的位置寄存器指令。
 - 向外围设备发送信号、以及从外围设备接收信号的 I/O（输入/输出）指令。
 - 所定义的条件成立时，改变程序流向的转移指令（IF, WITCH, CALL）。
 - 使程序执行推迟的等待指令。
 - 添加到程序上的注解。
 - 其他指令。
- 程序末尾记号表示该程序中再也没有其他指令的记号。

3.1 程序属性

程序属性的设定，在程序属性画面上进行。在程序一览画面上选择“程序属性”后予以显示。

3.1.1 程序名

使用程序名来区别存储在控制装置内的存储器中的程序。在相同控制装置内不能创建相同名称的程序。

长度

程序名的长度为 1 ~ 60 个字符。程序名相对程序必须是独一无二的。

可以使用的字符

字母：英文字母。

数字：0 ~ 9。程序名不可从数字开始。

记号：仅限_（下划线）。不可使用@（@符号）和*（星号）。

内容

程序必须以能够弄清其目的和功能的方式进行命名。

3.1.2 程序备注

创建新的程序时，还可以在程序名上添加程序备注。程序备注用来记述希望在程序列表界面上与程序名一起显示的附加信息。

长度

程序注解的长度为 1 ~ 1000 个字符。

可以使用的字符

可使用任何字符、数字、符号以及汉字。

内容

程序备注必须以能够弄清程序的目的和功能的方式进行描述。

3.1.3 程序类型

有如下所示的程序类型：

- Main - 指定可作为主程序而从示教器或外部装置启动的程序。
- None - 指定作为一般程序，可直接执行或从工作程序中呼叫并执行特定作业的程序。

Main 主程序

可以被本地启动、主程序号启动、主程序号简易模式启动。当程序类型选择 Main 时，程序属性界面会增加程序起始点和程序号的列表，并且程序员权限可以选择是否禁用“程序起始点”功能。如图 3.5 所示。只有当程序类型选择为 Main 时，需要设置程序号。程序起始点详情见 4.3 节；程序号详情见 4.2.2 节的补充说明部分。

程序属性

程序适配机型: GBT-S10A-600✕

程序名称	<input type="text" value="A"/>
程序备注	<input type="text" value="程序备注支持不超过1000个字节的内容。"/>
程序类型	<input type="text" value="Main"/>
程序起始点	<input checked="" type="checkbox" value="启用"/> <input type="checkbox" value="未设置"/>
程序号	<input type="text" value="1"/> <input type="text" value="0b00000001"/>

当前系统未创建Home-Pose，请至Home-Pose页添加。

创建时间: admin 创建于 2025-08-12 15:44:23
修改时间: admin 修改于 2025-08-12 15:44:23

图 3.4 程序号设置

程序属性

程序适配机型: GBT-S10A-600

程序名称	<input type="text" value="A"/>
程序备注	<input type="text" value="程序备注支持不超过1000个字节的内容。"/>
程序类型	<input type="text" value="Main"/>
程序起始点	<input type="button" value="启用"/> <input type="button" value="未设置"/> <small>当前系统未创建Home-Pose, 请至Home-Pose页添加。</small>
程序号	<input type="text" value="1"/> <input type="text" value="0b00000001"/>

创建时间: admin 创建于 2025-08-12 15:44:23
修改时间: admin 修改于 2025-08-12 15:44:23

图 3.5 程序起始点设置

None 一般程序:

没有什么特殊限制或特殊性。在自动模式时, 无法通过除了本地启动方式(手动按启动按钮)以外的方式启动起来。但是可以被其他程序调用。

3.1.4 程序保护

可以通过写保护来指定是否可以更改及删除程序。

程序保护开启的情况下, 不能将数据追加到程序中, 或修改程序。在结束程序调试工作后, 为避免自己或其他人员改写程序, 在程序列表界面开启程序保护 (图 3.2 程序列表界面)。



图 3.2 程序列表界面

3.2 行号码、程序末尾记号、箭头和参数

行号码

行号码自动插入到程序上所追加的各指令旁。在删除指令时，程序将自动地重新赋予号码，使得最初一行始终为行 1，第 2 行为行 2。

改变程序时，可以使用行号码通过光标来指定将哪一行作为移动、删除、范围指定的对象。

此外，还可以通过程序编辑画面中的“跳转到”，将光标移动到目标行号码。

程序末尾记号

程序末尾记号 (END)，自动显示在程序中的最后指令之后。随着新指令的追加，程序末尾记号可在将该位置保持在程序最后一行的同时，朝向画面下方移动。

程序在执行程序最后的指令到末尾记号时，程序结束，箭头停在末尾记号处。

箭头

箭头指向的程序行号码，表示程序执行到该行

本章下面的内容，将就程序的创建 / 修改中所需的程序指令进行说明。

程序的创建 / 修改，在程序编辑画面上进行（有关程序的创建与修改，见 4.1 节）

参数 i

参数 i 是在控制指令（动作指令以外的程序指令）的指定中所使用的指数。参数有直接指定和间接指定之分。直接指定，通常情况下指定 1 ~ 32767 范围内的整数。值的范围，随所使用的指令种类而不同。间接指定用来指定寄存器号码。

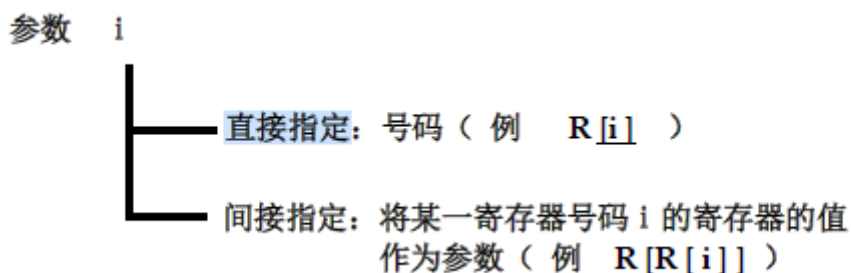


图 3.6 参数 i 的格式

3.3 动作指令

所谓动作指令，是指以指定的移动速度和移动方法使机器人向作业空间内的指定位置移动的指令。动作指令中指定的内容有如下一些。指令格式示如图 3.3 所示。

- 动作类型 - 指定向指定位置的轨迹控制。
- 位置数据 - 对机器人将要移动的位置进行示教。
- 移动速度 - 指定机器人的移动速度。
- 定位类型 - 指定是否在指定位置定位。
- 动作附加指令 - 指定在动作中执行附加指令。

要进行动作指令的示教（有关动作指令的创建，见 4.1.5 小节；有关动作指令的修改，见 4.1.6 小节）。

3.3.1 动作类型

动作类型指定向指定位置的移动轨迹。动作类型有：不进行轨迹控制/姿势控制的关节动作、进行轨迹控制/姿势控制的直线动作、圆弧动作。

- 关节动作（MOVEJ）
- 直线动作（MOVEL）
- 圆弧动作（MOVEC）
- 门型动作（只适用于四轴机器人）

关节动作（MOVEJ）

关节动作是将机器人移动到指定位置的基本的移动方法。机器人沿着所有轴同时加速，移动后，同时减速后停止。移动轨迹通常为非线性。在对结束点进行示教时记述动作类型。关节移动速度%（相对最大移动速度的百分比）。移动中的工具姿势不受

到控制。

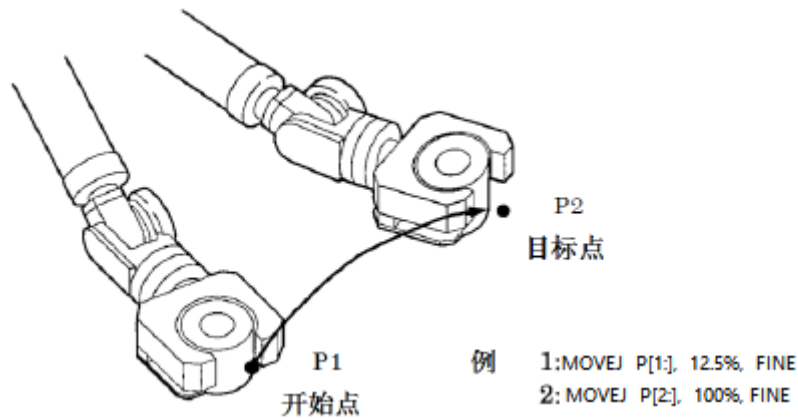


图 3.7 关节动作运动示意图

直线动作 (MOVEJ)

直线动作是以线性方式对从动作开始点到结束点的工具中心点移动轨迹进行控制的一种移动方法。在对结束点进行示教时记述动作类型。直线移动速度的单位是 mm/s。将开始点和目标点的姿势进行分割后对移动中的工具姿势进行控制。

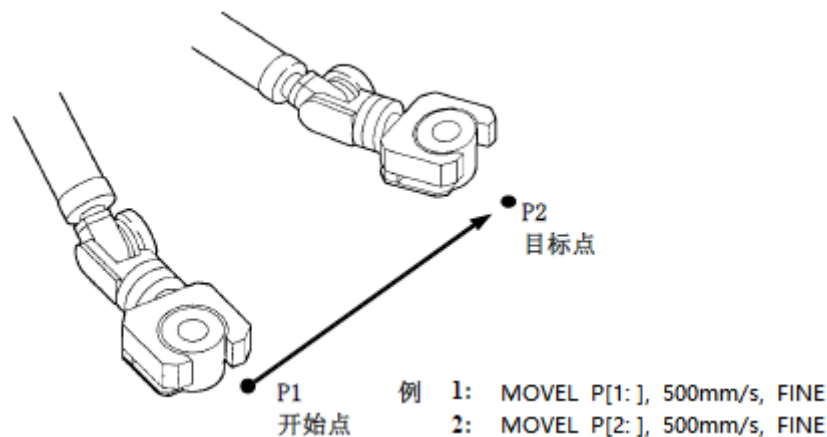


图 3.8 直线动作运动示意图

回转动作是使用直线动作，使工具的姿势从开始点到结束点以工具中心点为中心回转的一种移动方法。将开始点和目标点的姿势进行分割后对移动中的工具姿势进行控制。移动轨迹（工具中心点移动的情况下）通过线性方式进行控制。

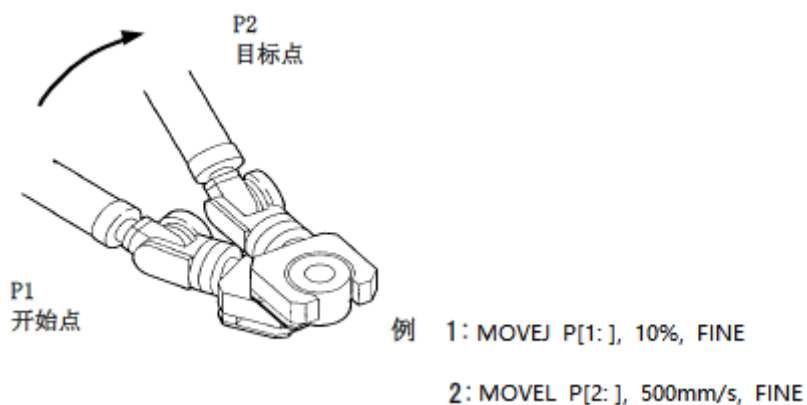


图 3.9 回转动作运动示意图

圆弧动作 (MOVEC)

圆弧动作是从动作开始点通过经过点到结束点以圆弧方式对工具中心点移动轨迹进行控制的一种移动方法。其在一个指令中对经由点和目标点进行示教。圆弧移动速度的单位是 mm/s。将开始点、经由点、目标点的姿势进行分割后对移动中的工具姿势进行控制

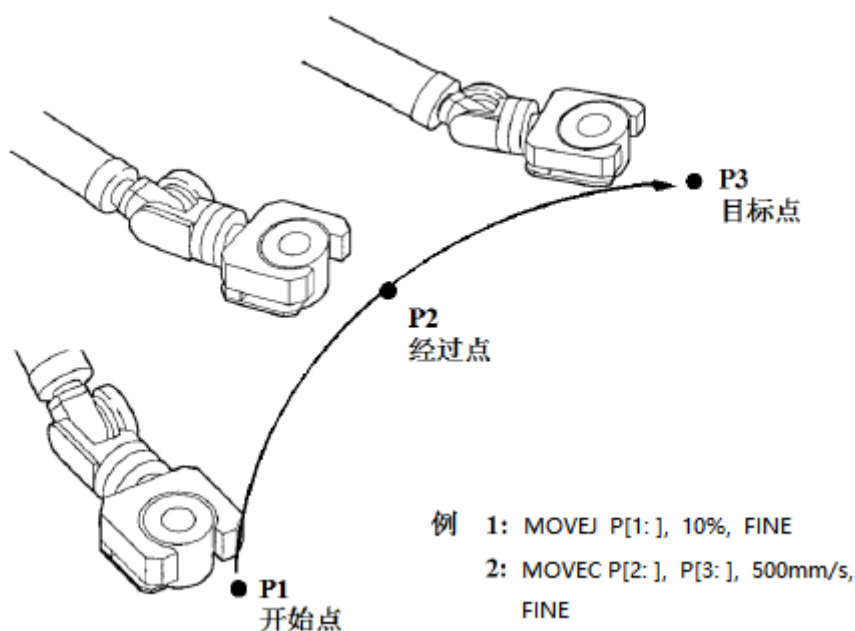


图 3.10 圆弧动作运动示意图

门型动作

门型动作主要运动方式是抬起 Z 轴到指定高度，然后移动轨迹到目标点正上方，再下降到目标点进行控制的一种移动方法。

门型动作 (JUMP)

JUMP 指令是指定机器人做门型运动的指令（首先垂直上升、然后水平移动，最后垂直下降的门型动作），该指令是将机器人当前点(连续自动运行时是指上一条指令的运动终点)作为 JUMP 指令起点，语句中指定的点作为 JUMP 指令终止点。全过程都是关节运动方式。

指令格式：

JUMP P[i], speed1, speed2, 定位类型, LimZ, 附加条件指令 1, 附加条件指令 2

语句参数说明：

- P[i]：表示目标位置；可直接由 P[i]指定，也可由 PR[i]间接指定
- Speed1:速度单位是 mm/s，可直接由常数指定，也可由 MR[i]间接指定，指上升下降的速度
- Speed2: 速度倍率值的范围为 1 ~ 100%，指水平移动的速度
- 定位类型：有 SD 与 FINE 两种，具体见 3.3.4 节
- LimZ：表示 Jump 运动中机器人的 Z 坐标上限值，可直接由常数指定，也可由 MR[i]间接指定
- 附加条件指令 1:可不选（省略）。指令是 Approach 一般形式为 ARCH (Arch1,Arch2)。Arch1：可以是 R 寄存器或数字。用于通过 Jump 命令指定水平动作前的转移距离（从出发点算起的垂直距离）。（单位：mm） Arch2：可以是 R 寄存器或数字。用于通过 Jump 命令指定完全结束水平移动阶段的接近距离（从目的点算起的垂直距离）。（单位：mm）
- 附件条件指令 2：可不选（省略）。指令有 Offset, Tool Offset, TB, TA, DB 可选（具体见 3.3.5 节）

指令编程示例：

```
JUMP P[1:], 500mm/s, 12.5%, SD20, LimZ 130, ARCH (30 50), Offset
```

门型动作 (JUMP3)

JUMP3 指令是指定机器人做门型运动的指令（用于组合 2 个直线动作与 1 个关节动作）该指令适用于 SCARA 机器人(4 轴机器人)。包含起始点和三个指定点，相邻两点依次组成 3 条线段。首尾线段采用直线运动，中间段采用 PTP 运动语句中指定的点作为 JUMP3 指令终止点。

指令格式：

JUMP3 P[i], P[j], P[k], speed1, speed2, 定位类型, 附加条件指令

语句参数说明:

- P[i]: 高于当前位置的转移点; 可直接由 P[i]指定, 也可由 PR[i]间接指定
- P[j]: 高于目标坐标点的接近点; 可直接由 P[i]指定, 也可由 PR[i]间接指定
- P[k]: 目标坐标点; 可直接由 P[i]指定, 也可由 PR[i]间接指定
- Speed1:速度单位是 mm/s, 可直接由常数指定, 也可由 MR[i]间接指定, 指 Z 轴抬升下降时的速度
- Speed2: 速度倍率值的范围为 1 ~ 100%, 指 P[i]点到 P[j]时的速度
- 定位类型: 有 SD 与 FINE 两种, 具体见 3.3.4 节
- 附件条件指令: 可不选 (省略)。指令有 Offset, Tool Offset, TB, TA, DB 可选 (具体见 3.3.5 节)

指令编程示例:

```
JUMP3 P[1: ], P[2: ], P[3: ], 500mm/s, 12.5%, SD20, Offset
```

门型动作 (JUMP3CP)

JUMP3CP 指令是指定机器人做门型运动的指令 (用于组合 3 个 直线动作) 该指令适用于 SCARA 机器人 (4 轴机器人)。包含起始点和三个指定点, 相邻两点依次组成 3 条线段, 3 条线段都是直线。

指令格式:

```
JUMP3CP3 P[i], P[j], P[k], speed, 定位类型, 附加条件指令
```

语句参数说明:

- P[i]: 高于当前位置的转移点; 可直接由 P[i]指定, 也可由 PR[i]间接指定
- P[j]: 高于目标坐标点的接近点; 可直接由 P[i]指定, 也可由 PR[i]间接指定
- P[k]: 目标坐标点; 可直接由 P[i]指定, 也可由 PR[i]间接指定
- Speed:速度单位是 mm/s, 可直接由常数指定, 也可由 MR[i]间接指定, 指 Z 轴抬升下降时的速度
- 定位类型: 有 SD 与 FINE 两种, 具体见 3.3.4 节
- 附件条件指令: 可不选 (省略)。指令有 Offset, Tool Offset, TB, TA, DB 可选 (具体见 3.3.5 节)

指令编程示例:

```
JUMP3CP P[1: ], P[2: ], P[3: ], 500mm/s, SD20, Offset
```

3.3.2 位置资料

位置资料存储机器人的位置和姿势。在对动作指令进行示教时，位置资料同时被写入程序。

位置资料有：基于关节坐标系的关节坐标值、和通过作业空间内的工具位置和姿势来表示的直角坐标值。

直角坐标值

基于直角坐标值的位置资料，通过 4 个要素来定义。直角坐标系中的工具中心点（工具坐标系原点）位置、工具姿势、形态、所使用的直角坐标系。

直角坐标系中使用世界坐标系或用户坐标系。有关这些直角坐标系的选择，将在本节稍后叙述。

位置和姿势

- 位置 (X, Y, Z)，以三维坐标值来表示直角坐标系中的工具中心点（工具坐标系原点）位置。
- 姿势 (A, B, C)，以直角坐标系中的 X, Y, Z 轴周围的回转角来表示。

形态

形态是指机器人主体部分的姿势。有多个满足直角坐标值 (X,Y,Z,A,B,C) 条件的形态。要确定形态，需要指定每个轴的关节配置和回转数。

关节配置

关节配置表示手腕和手臂的配置。指定手腕和手臂的控制点相对控制面位于哪一侧。控制面上控制点相互重叠时，机器人位于奇异点（特殊姿势）。奇异点上由于存在着无限种基于指定直角坐标值的形态，机器人不能操作。

- 机器人不能在位于奇异点的位置操作。（有的情况下，可选择最便于获取的形态进行操作。）这种情况下，可通过关节坐标值进行示教。
- 在直线/圆弧中，机器人不能通过路径上的奇异点（无法改变关节配置）。这种情况下，使用关节动作。

回转数

四轴机器人的回转数表示轴 (J4) 的回转数；六轴机器人的回转数表示轴 (J3、J4、J6) 的回转数。这些轴回转一周后返回相同位置，指定回转几周。在各轴处在 0° 的姿势下，回转数为 0。

在执行编程的直线、圆弧动作的情况下，机器人在选取离开始点的姿势最接近姿势的同时向目标点方向移动。此时，目标点的回转数将被自动选定，所以在目标点位

置的机器人实际回转数，在某些情况下会与所示教的位置资料的回转数不同。

核实直角坐标系

直角坐标系的核实，对再现基于直角坐标值的位置资料时使用哪个坐标系号码的直角坐标系进行检测。

在工具坐标系号码中指定了 0 ~ 10，在用户坐标系号码中指定了 0 ~ 10 的情况下，示教时使用的坐标系号码应与再现时的坐标系号码相同，否则可能会发生机器人碰撞。坐标系号码在位置示教时被写入位置资料。要改变被写入的坐标系号码，使用工具/用户，更换（激活）坐标功能后，重新示教该点位即可。

工具坐标系号码 (TF)

工具坐标系号码，由末端法兰坐标系或工具坐标系的坐标系号码指定。工具侧的坐标系由此而确定。

- 0：使用默认工具坐标系（末端法兰坐标系）。
- 1 ~ 10：使用所指定的工具坐标系号码的工具坐标系。

用户坐标系号码 (UF)

用户坐标系号码，由世界坐标系或用户坐标系的坐标系号码指定。作业空间的坐标系由此而确定。

- 0：使用基坐标系。
- 1 ~ 10：使用所指定的用户坐标系号码的用户坐标系。

位置变量 — P[i]

位置变量是标准的位置资料存储变量。在对动作指令进行示教时，自动记录位置资料。

在进行直角坐标值的示教时，使用如下直角坐标系和坐标系号码。

- 当前所选的工具坐标系号码的坐标系 (UT = 0 ~ 10)。
- 当前所选的用户坐标系号码的坐标系 (UF = 0 ~ 10)。

再现时使用如下直角坐标系和坐标系号码。

- 所示教的工具坐标系号码的坐标系 (UT = 0 ~ 10)。
- 所示教的用户坐标系号码的坐标系 (UF = 0 ~ 10)。

位置寄存器 — PR[i]

位置寄存器是用来存储位置资料的通用的存储变量。（有关位置寄存器的位置示教，见 5.1.3 节）

在进行直角坐标值的示教时，使用如下直角坐标系和坐标系号码。

- 当前所选的工具坐标系号码的坐标系（UT = 0 ~ 10）。
- 当前所选的用户坐标系号码的坐标系（UF = 0 ~ 10）。

再现时使用如下直角坐标系和坐标系号码。

- 所示教的工具坐标系号码的坐标系（UT = 0 ~ 10）。
- 所示教的用户坐标系号码的坐标系（UF = 0 ~ 10）。



由于 PR 寄存器不带 UF/TF 信息，程序执行时表示指定 TF 在当前激活使用 UF 下的位姿信息；所以切换不同的 UF/TF，实际机器人在空间中位置可能不同。

P 表示局部位姿数据，包含 UF/TF 信息，如果运动指令使用 P 数据时，UF_No.、TF_No.与 P 数据记录不一致，则程序无法继续向下执行，需要在该运动指令前修改 UF_No.、TF_No.使其与 P 数据记录一致。

位置号码 - P[i]

可为位置号码和位置寄存器号码添加名称，名称最多为 31 个字符。

例: MOVEJ P[1:point1], 10%, FINE

MOVEJ PR[1:point2], 10%, FINE

3.3.3 移动速度

在运动指令中指定机器人的移动速度。在程序执行时，移动速度受到速度倍率的限制。速度倍率值的范围为 1 ~ 100%。在移动速度中指定的单位，根据动作指令所示教的动作类型而不同。

MOVEJ P[1:], 10%, FINE

动作类型为关节动作的情况下，在 1 ~ 100 % 的范围内指定相对最大移动速度的比率。

MOVEJ P[2:], 500mm/s, FINE

动作类型为直线动作、圆弧动作情况下，单位为 mm/s，在 1 ~ 4000 mm/s 之间指定。

通过运动寄存器（MR）指定速度

可以通过运动寄存器来指定速度。由此，便可在运动寄存器中进行移动速度的计算后，指定动作指令的移动速度。



本功能可以通过运动寄存器来自由改变机器人的移动速度。因此，根据所指定的运动寄存器值，有时会导致机器人在意想不到的速度下动作。使用本功能的情况下，应注意在示教时 / 运转时，对所指定的运动寄存器值进行充分确认。

通过运动寄存器指定的移动速度的动作指令显示形式

- MOVEJ P[1:], MR[i:]%, FINE
- MOVEL P[2:], MR[i:]mm/s, FINE
- MOVEC P[3:], P[4:], MR[i:]mm/s, FINE

3.3.4 定位类型

根据定位类型，定义动作指令中的机器人的动作结束方法。标准情况下，定位类型有两种。

- FINE 定位类型
- SD 定位类型

FINE 定位类型

MOVEJ P[1:], 10%, FINE

根据 FINE 定位类型，机器人在目标位置停止（定位）后，向着下一个目标位置移动。

SD 定位类型

MOVEJ P[1:], 10%, SD50

根据 SD 定位类型，机器人靠近目标位置，但是不在该位置完全停止就进行下一个位置的动作。

机器人靠近目标位置到什么程度，由 0 ~ 1000 之间的值来定义。

指定 0 时，机器人在最靠近目标位置处动作，但是不在该位置完全停止就进行下一个位置的动作。

SD 的值越大，机器人在目标位置附近减速越小，并且离目标位置越远。



在指定了 SD 的动作指令后编程等待等指令的情况下，标准设定下机器人会在拐角部分的轨迹上停止，执行该指令。

在 SD 方式下连续执行距离短而速度快的多个动作的情况下，即使 SD 的值为 1000，也会导致机器人减速。

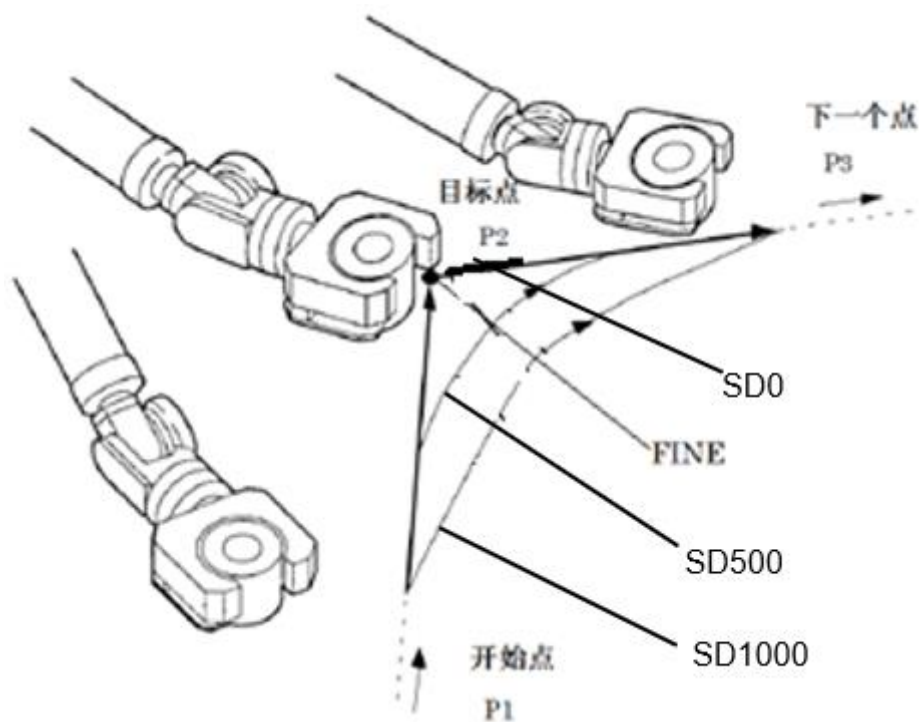


图 3.11 转角半径效果示意图

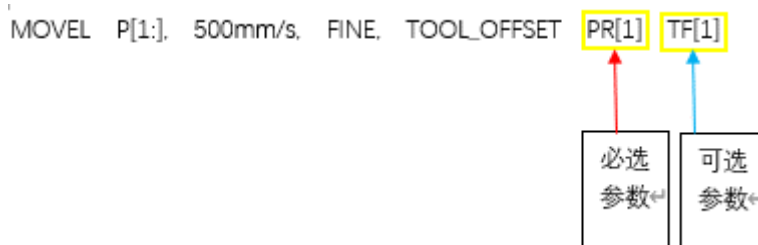
3.3.5 动作附加指令

动作附加指令是在机器人动作中使其执行特定作业的指令。动作附加指令有如下列表：

章节序号	附加指令	指令语句
3.3.5.1	工具补偿指令	Tool_Offset
3.3.5.2	位置补偿指令	Offset
3.3.5.3	提前执行指令	TB
3.3.5.4	延后执行指令	TA

3.3.5.5	指定距离触发指令	DB
3.3.5.6	切换运动参数指令	Swift
3.3.5.7	远程坐标系指令	RTCP
3.3.5.8	跳转指令	SKIP
3.3.5.9	加速度比例系数	ACC

3.3.5.1 工具补偿指令 (Tool_Offset)



必选参数：表示偏移量，类型是 PR 寄存器

可选参数：作为偏移基准的工具坐标系

工具补偿指令，在位置资料中所记录的目标位置，使机器人移动到仅偏移工具补偿条件中所指定的补偿量后的位置。偏移的条件，由工具补偿条件指令来指定。

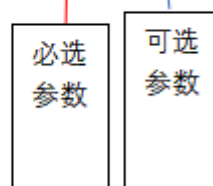
工具补偿条件指定如下要素：

- 位置寄存器指定偏移的方向和偏移量。
- 偏移时使用工具坐标系。
- 在没有指定作为偏移基准的工具坐标系的情况下，使用当前所使用的工具坐标系。若指定作为偏移基准的工具坐标系，则以指定的坐标系为基准进行偏移。

如果基础运动指令是 MoveJ（被偏移的对象点即 Move 指令中的目标 P[n]是以关节形式被执行的），此时即使指定偏移量 PR[x]的表达式是直角坐标系，也将被转换成关节坐标再去执行关节上的偏移，但转换过程中是基于该 TF [i]参数指定的偏移基准执行的。

3.3.5.2 位置补偿指令 (Offset)

MOVE P[1:], 500mm/s, FINE, OFFSET PR[1] UF[1]



必选参数：表示偏移量，类型是 PR 寄存器

可选参数：作为偏移基准的用户坐标系

位置补偿指令，在位置资料中所记录的目标位置，使机器人移动到仅偏移位置补偿条件中所指定的补偿量后的位置。偏移的条件，由位置补偿条件指令来指定。

位置补偿条件指定如下要素。

- 位置寄存器指定偏移的方向和偏移量。
- 位置资料为关节坐标值的情况下，使用关节的偏移量。
- 位置资料为直角坐标值的情况下，使用可选参数指定的用户坐标系为基准进行偏移，没有指定的情况下，使用当前所选的用户坐标系（UF）

在位置补偿指令中必选参数除了可以使用 PR 还可以使用一个具体的点位信息：C_VEC (X,Y,Z,A,B,C) 或 J_VEC (J1,J2,J3,J4,J5,J6) 直接指定偏移量;其中 C_VEC 中的数据是笛卡尔类型，J_VEC 中的数据是关节类型。

如果基础运动指令是 MoveJ（被偏移的对象点即 Move 指令中的目标 P[n]是以关节形式被执行的），此时即使指定偏移量 PR[x]的表达式是直角坐标系，也将被转换成关节坐标再去执行关节上的偏移，但转换过程中是基于该 UF []参数指定的偏移基准执行的。

MOVE 运动指令仅在执行的瞬间读取 PR 寄存器的值，后续机器人运动期间不影响 PR 寄存器的读写。

3.3.5.3 提前执行指令 (TB)

在机器人动作结束的指定时间之前，进行信号输出、调用子程序（不含运动指令，若子程序包含运动指令会报错，错误代码为“Program-2328“），可指定的时间范围为 0.01-30s，若输入小于 0.01 的数不会保存成功（0 除外）。

格式：

- 动作语句 + TB + 指定时间 + CALL 程序名
- 动作语句 + TB + 指定时间 + 输出信号

输出信号可以选择的有 DO[i]

例子:

- MoveJ P[1] 100% FINE TB 1.00S CALL PROGRAM
机器人以关节运动方式移动到 P1 点, 在到达 P1 点的前 1s,调用程序 PROGRAM
- MoveJ P[1] 100% FINE TB 1.00S DO[1]=ON
机器人以关节运动方式移动到 P1 点, 在到达 P1 点的前 1s,把 DO[1]置为 ON

3.3.5.4 延迟执行指令 (TA)

在机器人动作结束的指定时间之后, 进行信号输出、调用子程序 (不含运动指令, 若子程序包含运动指令会报错, 错误代码为“Program-2328”), 可指定的时间范围为 0.01-1s, 若输入小于 0.01 的数不会保存成功 (0 除外)。

格式:

- 动作语句 + TA + 指定时间 + CALL 程序名
- 动作语句 + TA + 指定时间 +输出信号

输出信号可以选择的有 DO[i]

例子:

- MoveJ P[1] 100% FINE TA 1.00S CALL PROGRAM
机器人以关节运动方式移动到 P1 点, 在到达 P1 点的后 1s,调用程序 PROGRAM
- MoveJ P[1] 100% FINE TA 1.00S DO[1]=ON
机器人以关节运动方式移动到 P1 点, 在到达 P1 点的后 1s,把 DO[1]置为 ON

3.3.5.5 指定距离触发指令 (DB)

当机器人 TCP 运动到与运动指令目标位置的距离在指定距离以内时, 使用该指令可让机器人在动作的同时还能执行信号输出或调用子程序 (不含运动指令)。

格式:

- 动作指令 + DB 距离指定值 + CALL 程序名
- 动作指令 + DB 距离指定值 + 信号输出

输出信号可以选择的有 DO[i]

参数:

- 距离指定值: TCP 进入以运动指令目标点为中心的球形范围内时, 执行触发指令。该球形区域的半径由距离指定值确定。

例子:

- MOVEL P[1] 100% FINE DB 100.00mm, DO[3]=ON

机器人以直线运动方式向 P1 点移动, 在机器人的 TCP 进入到以 P1 点为球心, 100mm 为半径的空间内时, 把 DO[3]置为 ON

- MOVEL P[2] 100% FINE DB 50.00mm, CALL A

机器人以直线运动方式向 P1 点移动, 在机器人的 TCP 进入到以 P1 点为球心, 100mm 为半径的空间内时, 调用程序 A

3.3.5.6 切换运动参数指令 (Swift)

SWIFT 换挡指令: 用于切换运动参数 (加速度, 加加速度等), 以满足更快的节拍要求。

例子:

- MOVEJ P[1] 50% FINE, SWIFT

3.3.5.7 远程坐标系指令 (RTCP)

在执行远程 TCP 动作时, 需在指令中添加远程工具动作附加指令 RTCP。RTCP 坐标系与用户坐标系本质是一样的, 数据也是一样的, 仅当在运动指令中添加了 RTCP 附加指令时, 记录点位时所使用的用户坐标系就会变成 RTCP 坐标系, 若不带 RTCP 附加指令, 则依旧保持为用户坐标系。该变化体现在控制系统中的算法处理上, 在配置页面上不做 RTCP 坐标系与用户坐标系的明显区分。



RTCP 不能在关节动作下使用。

例子:

相对远程 TCP, 在 2000mm/s 的相对速度下将工件移动到 P[1]点:

```
MoveL P[1] 2000mm/s Fine RTCP
```

相对远程 TCP, 在 2000mm/s 的相对速度下使工件经由 P[1]移动到 P[2]:

```
MoveC P[1] P[2] 2000mm/s Fine RTCP
```

3.3.5.8 条件跳转指令 (SKIP)

SKIP 指令用于当满足 SKIP CONDITION 指令（见 3.6.5 节）设置的跳转条件时，从含有 SKIP 指令的当前行直接跳转到目标指令行或目标程序，常与动指令 SKIP CONDITION 搭配使用。

指格式：动作指令 + SKIP GOTO LABEL[i] / CALL + 程序名

例子：

```
1 SKIP CONDITION DO[2]=ON
2 MOVEJ P[1] , 200 mm/s , FINE , SKIP GOTO LABEL[1]
3 MOVEJ P[2] , 12.5% , FINE
4 LABEL[1]
5 MOVEJ P[3] , 12.5% , FINE
```

程序从第 1 行开始执行，当 DO[2]=ON 时，执行程序第 2 行立刻跳转到第 4 行，从第 4 行继续向下执行，当 DO[2]=OFF 时，程序第 2 行执行结束后从当前行继续向下顺序执行。

3.3.5.9 加速度倍率指令 (ACC)

ACC 仅作用于单条运动指令，当运动指令未使用 ACC 附加指令时，使用全局加速度比例系数进行运动规划。当运动指令使用 ACC 附加指令时，则以 ACC 附加指令作为加速度比例系数进行规划。

格式：ACC 80%

参数：

- 加速度比例：加速度比例百分比，10% 至 100% (修改此数值仅可由厂商权限进行!)。

例子：

```
MOVEJ P[1], 500 mm/s , FINE , ACC 80%
```

3.4 寄存器指令

寄存器指令是进行寄存器的算术运算的指令。寄存器有如下几种。

- 数值寄存器指令
- 位置寄存器指令
- 位置寄存器要素指令
- 字符串寄存器、字符串指令

- 运动寄存器指令
- Modbus 专用寄存器指令
- 码垛寄存器指令



所有的寄存器都可以直接通过数值来指定调用，或者间接通过数值寄存器来组合运算调用。称为直接指定法和间接指定法。

表达式范例如下：PR[R[20]]、SR[20+R[2]]、R[R[1]+R[2]]

寄存器运算，可以进行如下所示的多项式运算。

例

$R[2]=R[3]-R[4]+R[5]-R[6]$

$R[10]=R[2]*100/R[6]$

3.4.1 数值寄存器指令

数值寄存器用来存储某一整数或小数值得变量。标准情况下提供有 1500 个数值寄存器。

$R[i] = (\text{值})$ ，其中 i 为数值寄存器的编号 (1-1500)

值可以是：

- 常数
- $R[i]$: 寄存器 $R[i]$ 的值
- $PR[i, j]$: 位置 PR 要素 $[i, j]$ 的值
- $DI[i]$: 数字输入信号
- $DO[i]$: 数字输出信号
- $UI[i]$: 系统输入信号
- $UO[i]$: 系统输出信号
- 字符串
- 方法：SIN、COS 等 (方法具体见 3.10 节)

数值寄存器的算术符号有加 (+)、减 (-)、乘 (*)、除 (/)、取余数 (MOD)、取商的整数值(DIV)

举例如下：

$$R[i] = (\text{值}) + (\text{值})$$

$R[i] = (\text{值}) + (\text{值})$ 指令，将 2 个值的和代入数值寄存器。

$$R[i] = (\text{值}) - (\text{值})$$

$R[i] = (\text{值}) - (\text{值})$ 指令，将 2 个值的差代入数值寄存器。

$$R[i] = (\text{值}) * (\text{值})$$

$R[i] = (\text{值}) * (\text{值})$ 指令，将 2 个值的积代入数值寄存器。

$$R[i] = (\text{值}) / (\text{值})$$

$R[i] = (\text{值}) / (\text{值})$ 指令，将 2 个值的商代入数值寄存器。

$$R[i] = (\text{值}) \text{ MOD } (\text{值})$$

$R[i] = (\text{值}) \text{ MOD } (\text{值})$ 指令，将 2 个值的余数代入数值寄存器。

$$R[i] = (\text{值}) \text{ DIV } (\text{值})$$

$R[i] = (\text{值}) \text{ DIV } (\text{值})$ 指令，将 2 个值的商的整数部分代入数值寄存器。

$$R[i] = (x - (x \text{ MOD } y)) / y$$

3.4.2 位置寄存器指令

位置寄存器指令，是进行位置寄存器的算术运算的指令。位置寄存器指令可进行代入、加法、减法处理。

位置寄存器，是用来存储位置资料 (X,Y,Z,A,B,C) 的变量。标准情况下提供有 200 个位置寄存器。

PR[i] = (值)

$PR[i] = (\text{值})$ 指令，将位置资料代入位置寄存器。其中 i 为位置寄存器的编号 (1-200)

值可以是：

- $PR [i]$ ：位置寄存器[i]的值
- $P [i]$ ：程序内的示教位置[i]的值
- L_POS ：当前位置的直角坐标值（见 3.8.4 节）
- J_POS ：当前位置的关节坐标值（见 3.8.3 节）

举例如下：

PR [1] = L_POS

PR [2] = PR [3]

PR [3] = P[1:]

PR[4] = PR[R[1]]

3.4.3 位置寄存器要素指令

位置寄存器要素指令，是进行位置寄存器的算术运算的指令。

PR[i,j]的 i 表示位置寄存器号码， j 表示位置寄存器的要素号码。位置寄存器要素指令可进行代入、加法、减法处理，以与数值寄存器指令相同的方式记述。

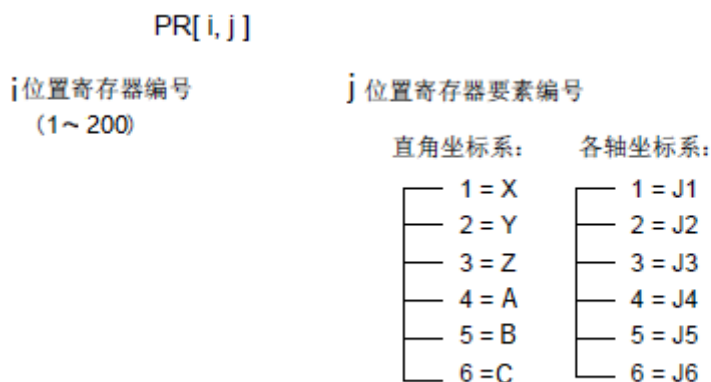


图 3.11 位置寄存器要素示意图

PR[i,j] = (值)

PR[i,j] = (值) 指令，将位置资料的要素值代入位置寄存器要素。

其中 i 为位姿寄存器的编号 (1-200)

值可以是：

- 常数
- R[i]: 寄存器[i]的值
- PR[i, j]: 位置 PR 要素[i, j]的值
- MH [i]: Modbus 寄存器 (保持型)
- MI [i]: Modbus 寄存器 (输入型)

Modbus 寄存器参考 3.4.6 节

举例如下：

PR [1, 1] = R [3]

PR [4, 3] = 324.5

PR[i,j] = (值) + (值)

PR[i,j] = (值) + (值) 指令，将 2 个值的和代入位置寄存器要素。

PR[i,j] = (值) - (值)

PR[i,j] = (值) - (值) 指令，将 2 个值得差代入位置寄存器要素。

PR[i,j] = (值) * (值)

PR[i,j] = (值) * (值) 指令，将 2 个值的积代入位置寄存器要素。

PR[i,j] = (值) / (值)

PR[i,j] = (值) / (值) 指令，将 2 个值的商代入位置寄存器要素。

PR[i,j] = (值) MOD (值)

PR[i,j] = (值) MOD (值) 指令，将 2 个值的余数代入位置寄存器要素。

PR[i,j] = (值) DIV (值)

PR[i,j] = (值) DIV (值) 指令，将 2 个值的商的整数部分代入位置寄存器要素。

PR [4, 3] = PR [1, 3] - 3.528

3.4.4 字符串寄存器、字符串指令

字符串寄存器，各自的寄存器中，最多可以存储 255 个字符。字符串寄存器数最多为 300 个。

SR[i] = (值)

SR[i]= (值) 指令，将字符串资料代入字符串寄存器。

值可以是：常数、(数值寄存器 R[i],字符串寄存器 SR[i])、字符串、方法

可从数值数据变换为字符串数据。保留小数点后 6 位，超多的部分四舍五入。

可从字符串数据变换为数值数据。变换值为字符串中最初出现字符前存在的数值。

例 SR[i] = R[j]

R [j] 的值	SR [i] 的结果
------------	--------------

$R [j] = 1234$	$SR [i] = "1234"$
$R [j] = 12.34$	$SR [i] = "12.34"$
$R [j] = 5.123456789$	$SR [i] = " 5.123457"$

例 $R[i] = SR[j]$

SR [j] 的值	R [i] 的结果
SR [j] = '1234'	R [i] = 1234
SR [j] = '12.34'	R [i] = 12.34

SR[i] = (值) (算符) (值)

SR[i] = (值) (算符) (值) 指令，将 2 个值结合起来，并将该运算结果代入字符串寄存器。

数据类型在各运算中，依赖于 (算符) 左侧的 (值)。

左侧的 (值) 若是字符串数据，则将字符串相互结合起来。

左侧的 (值) 若是数值数据，则进行算术运算。此时，右侧的 (值) 若是字符串，最初出现字符之前的数值用于运算。

算符是：+ (结合)

例 $SR[i] = R[j] + SR[k]$

R [j]、R [k] 的值	SR [i] 的结果
$R [j] = 123.456 + SR [k] = '345.678'$	$SR [i] = '456.134'$
$R [j] = 456 + SR [k] = '1abc2'$	$SR [i] = '457'$

例 $SR[i] = SR[j] + R[k]$

SR [j]、R [k] 的值	SR [i] 的结果
---------------------	--------------

SR [j] ='123.'+R [k] =456	SR [i] ='123.456'
SR [j] ='def'+R [k] =81573	SR [i] ='def81573'

3.4.5 运动寄存器指令

运动寄存器可以理解成运动指令专用的运动变量。方便调试大量相同的运动指令速度，MR 寄存器计算操作与数值寄存器 R 相同，MR 寄存器只能运用在运动指令中、仅在运动指令中出现。MR 寄存器仅支持整数型，如计算赋值出现了小数则舍尾取整。

MR[i] = 值

值的数据类型有常数、寄存器（R/PR[i,j]/MI[i]/MH[i]）两种

举例如下：

MR[1:]=500

MOVEL P[1:], MR[1:]mm/s, FINE

机器人以速度为 500mm/s 直线运动方式向 P1 点移动

3.4.6 Modbus 专用寄存器指令

Modbus 专用寄存器是在 Modbus 通讯中用来与 Modbus 主站进行通讯的专用寄存器。

MH/I[i] = (值)，其中 i 为 Modbus 专用寄存器的编号

值可以是：

- 常数
- R[i]: 寄存器 R[i] 的值
- PR[i,j]: 位置 PR 要素[i,j] 的值
- MI[i]: 输入型寄存器的值
- MH[i]: 保持型寄存器的值

3.4.7 码垛寄存器指令

码垛寄存器指令是用于直接读取、修改、计算码垛寄存器数值的指令。其具体应用方式如下。

3.4.7.1 码垛寄存器指令

已经建立成功并且配置完成通过校验的码垛工艺工程才可索引到具体的 PL 寄存器；

赋值形式：

$$PL[i] = [R,C,L] \quad \text{或} \quad PL[i] = PL[j]$$

3.4.7.2 码垛寄存器要素指令

用于直接读取、修改、计算码垛寄存器中某个要素数值的指令。其具体应用方式如下。

指令形式：

$$PL[i, R]$$

赋值形式：

$$PL[i, R] = MR[i] \text{或正整数}$$

码垛寄存器要素，值为 R（行）、C（列）、L（层），正整数，最大值视系统支持情况而定（使用时受限于具体码垛配置的行列层值，取值范围为[1, 255]）。

举例：

$$MR[2] = 3$$

$$PL[1, R] = MR[2]$$

$$PL[1, C] = MR[2]$$

$$PL[1, L] = 5$$

3.4.7.3 码垛寄存器用作条件判断

为便于使用者根据实际应用场景进一步细化程序逻辑控制，使符合生产工艺要求，用于程序中控制基于码垛寄存器的码垛堆积使用，码垛寄存器可用作条件判断：

例 1：

当行列层值为 1、2、3 时，执行...

$$IF PL[1] = [1,2,3]$$

...

例 2：

当第 1 行，偶数层时，执行...

...

```
ELSE IF PL[1] = [1,2-0,*]
```

```
...
```

条件判断中，[R,C,L]中的元素的表现形式与“设定路径条件的表述”的形式一致，不可出现加减乘除运算，不支持寄存器操作。例如以下情况都是不允许的：[2+3, *, *], [2/3, *, *], [R[1], *, *], [MR[1]-1, *, *]等类型。

3.5 IO 指令

IO 指令，通过读取输入信号(DI/UI/RI)状态及改变输出信号 (DO/RO) 状态，可以实现与外围设备如 PLC 的信号交互，控制应用流程。



I/O 信号，在使用前需要将逻辑号码映射给物理号码。(有关 I/O 的映射，见 2.1.1 节)

3.5.1 数字 I/O 指令

数字输入 (DI) 和数字输出 (DO)，是用户可以控制的输入/输出信号。

R[i] = DI[i]

R[i] = DI[i]指令，将数字输入的状态 (ON = 1、OFF = 0) 存储到寄存器中。

举例如下：

```
R [1] = DI [1]
```

```
R [R [3]] = DI [R [4]]
```

DO[i] = ON/OFF

DO[i] = ON / OFF 指令，接通或断开所指定的数字输出信号。

举例如下：

```
DO [1] = ON
```

```
DO [R [3]] = OFF
```

DO[i] = PULSE,[时间]

DO[i] = PULSE,[时间]指令，仅在所指定的时间内接通而输出所指定的数字输出；时间单位时秒。脉冲输出时间设置大于 0，时间小于 0 将报警。

举例如下：

```
DO [2] = PULSE , 0.2 s
```

DO [R [3]] = PULSE , 1.2 s

DO [i] = R[i]

DO[i] = R[i]指令，根据所指定的寄存器的值，接通或断开所指定的数字输出信号。若寄存器的值小于等于 0 就断开，大于 0 就接通

举例如下：

DO [1] = R [2]

DO [R [5]] = R [R [1]]

3.5.2 组 I/O 指令

组输入(GI)以及组输出(GO)信号，对几个数字输入/输出信号进行分组，以一个指令来控制这些信号。

R[i] = G[i]

R[i]=GI[i]指令，将所指定组输入信号的二进制值转换为十进制数的值代入所指定的寄存器。

举例如下：

R[1]= GI[1]

R [R[3]]=GI[R[4]]

GO[i]=(值)

GO[i]=(值)指令，将经过二进制变换后的值输出到指定的群组输出中。

举例如下：

GO [1]=0

GO [R [3]]=32767

GO[i]=R [i]

GO[i]=R[i]指令，将所指定寄存器的值经过二进制变换后输出到指定的组输出中。

举例如下：

GO [1]=R [2]

GO [R [5]]=R [R [1]]

3.5.3 机器人 I/O 指令

机器人 (RO/RI) I/O 指令与数字 (DO/DI) I/O 指令用法相同，参考数字 (DO/DI) I/O 指令用法即可

3.6 逻辑指令

捷勃特机器人逻辑指令包含 IF 指令、SWITCH 指令、WHILE 指令；逻辑指令也可理解为条件控制指令，通过一条或多条语句的执行结果（True 或者 False）来决定执行的代码块。可以通过下图来简单了解条件语句的执行过程：

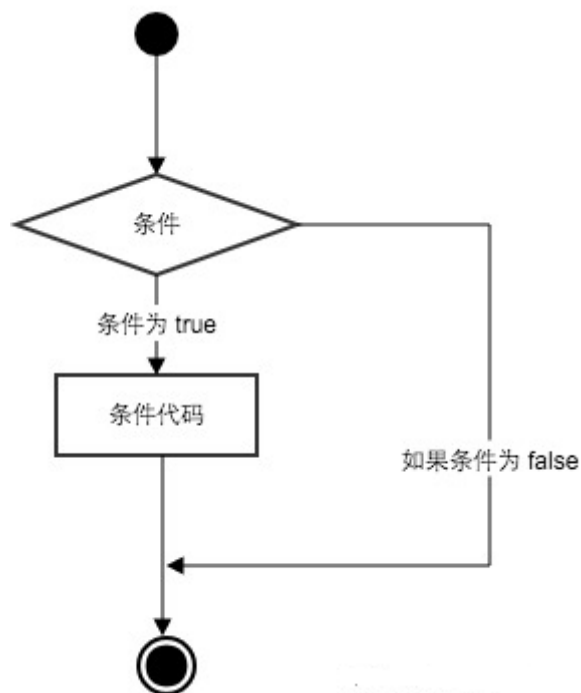


图 3.12 条件语句执行过程示意图

3.6.1 IF 指令

指令格式：

IF 条件 1

语句块 1

ELSE IF 条件 2

语句块 2

ELSE

语句块 3

END IF

- 如果 条件 1 为 True 将执行 语句块 1
- 如果条件 1 为 False，将判断 条件 2

- 如果条件 2 为 True 将执行 语句块 2
- 如果 条件 2 为 False，将执行语句块 3

IF 可判断的条件类型

- 寄存器类型（数值寄存器 R[i]、字符串寄存器 SR[i]）
- IO 类型(DO/UO/DI/UI)

3.6.2 SWITCH 指令

SWITCH 指令通常用于处理一个判断变量有多种不同值时的情形，用于简化程序，增加程序的可读性

指令格式：

```
SWITCH subject
```

```
CASE pattern_1
```

```
Action_1
```

```
BREAK
```

```
CASE pattern_2
```

```
Action_2
```

```
BREAK
```

```
DEFAULT
```

```
Action_3
```

```
BREAK
```

```
BREAKEND SWITCH
```

当所有 CASE 都无法匹配时，则执行 DEFAULT 后面的语句

当 subject 与 pattern_x 的值相同时，就执行对应的 Action_x

Subject 可以是寄存器(R[*]、SR[*])、IO 类型、常数、方法

Pattern 可以是寄存器(R[*]、SR[*])、IO 类型、常数、方法、字符串

BREAK 语句则是跳出动作



CASE 必须要与 BREAK 搭配使用

Switch 逻辑指令编程示例如下图所示：

```
2 SWITCH R[10: default]
3 CASE 1
4   R[11: default] = R[10: default] + 1
5   BREAK
6 CASE 2
7   R[11: default] = R[10: default] + 2
8   BREAK
9 END SWITCH
```

图 3.13 Switch 编程样例

3.6.3 WHILE 指令

WHILE 指令又称循环指令，一般用于控制需要重复执行的程序或动作。

指令格式

WHILE 判断条件

执行语句 1

CONTINUE

执行语句 2

BREAK

执行语句 3

END WHILE

循环指令 WHILE 运行时，机器人运行至不满足判断条件或执行 BREAK 语句后，才跳出循环指令，执行 END WHILE 以后的指令。continue 语句与 break 语句的区别是，continue 语句只结束本次循环，而不是终止整个循环。break 语句则是结束整个循环过程，不在判断执行循环的条件是否成立。

WHILE 执行流程图如下：Condition

Code block

Condition is false

Condition is true

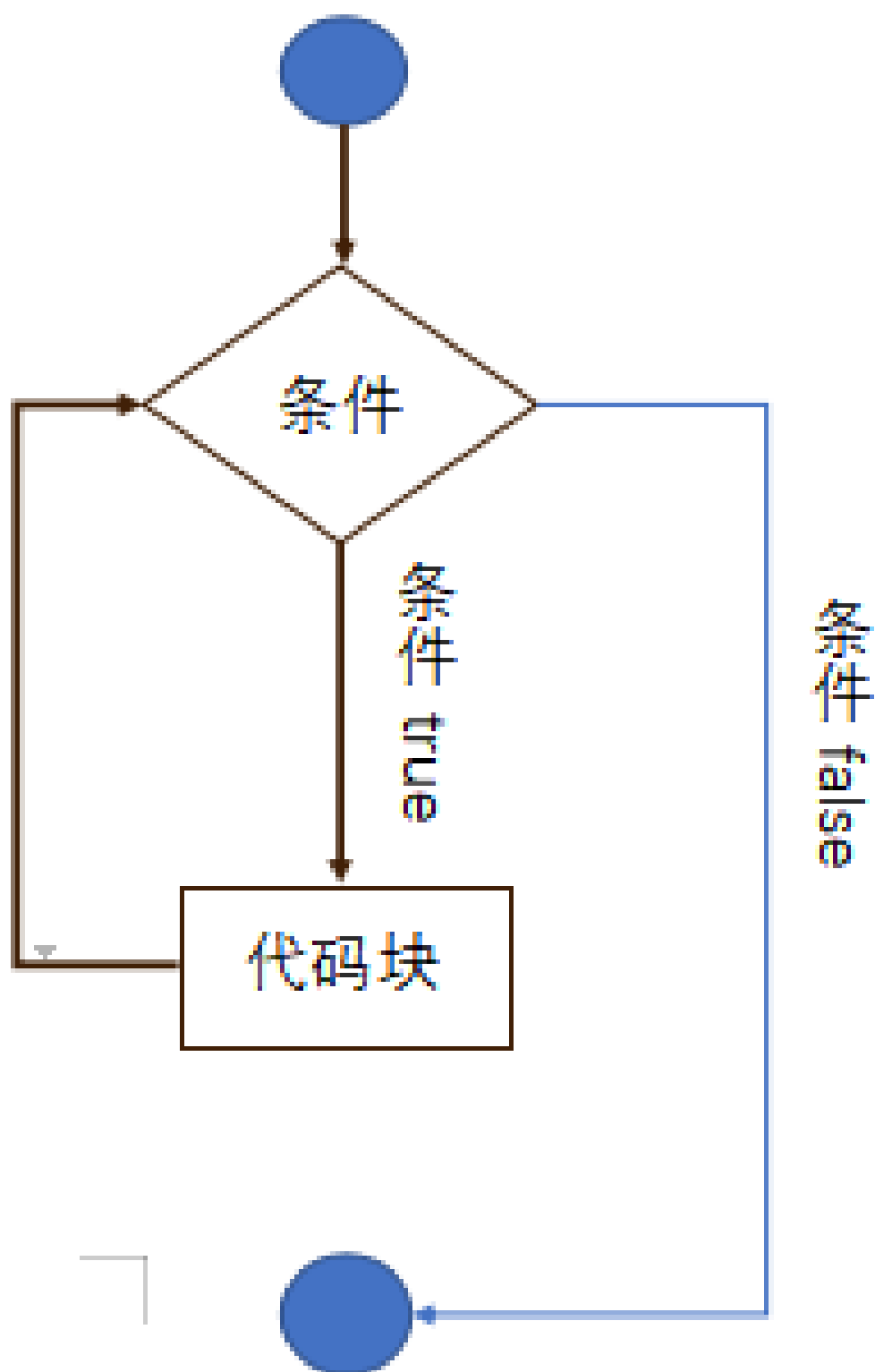


图 3.14 WHILE 语句执行流程

判断条件类型

- 寄存器类型（数值寄存器 R[i]、字符串寄存器 SR[i]）

- IO 类型(DO/UO/DI/UI)

WHILE 指令编程示例，如下图：

程序名：**WHILE** *内容被修改*

```
1 WHILE DI[2: ] = DI[2: ]
2   MOVEJ P[1: ] 100% FINE
3   WHILE DI[1: ] <> 1
4     MOVEJ P[2: ] 100% FINE
5     MOVEJ P[3: ] 4000mm/s FINE
6   END WHILE
7 END WHILE
8 END
```

图 3.15 WHILE 指令编程示例

3.6.4 GOTO 指令

GOTO 指令用于在同一程序内跳转程序到指定标签位置，并从指定标签位置开始继续向下执行程序；必须先插入标签指令 LABEL（见 3.8.11 节），再使用 GOTO 指令。

指令格式：

GOTO LABEL[i]

举例如下：

1 MOVEJ P[1] , 200 mm/s , FINE

2 LABEL[1]

3 MOVEJ P[1] , 200 mm/s , FINE

4 MOVEJ P[2] , 12.5% , FINE

5 GOTO LABEL[1]

6 MOVEJ P[3] , 12.5% , FINE

当程序从第 1 行执行到第 5 行（GOTO LABEL[1]）时，程序会跳转到第 2 行（LABEL[1]），从第 2 行继续往下执行。

3.6.5 SKIP CONDITION 指令

SKIP CONDITION 指令用于设置跳转条件，常与动作附件指令 SKIP（见 3.3.5 节）搭配使用。当满足跳转条件时，从含有 SKIP 指令的当前行直接跳转到目标指令行或目标程序。

指令格式：SKIP CONDITION 条件（IO 类型/寄存器类型）

举例如下：

```
1 SKIP CONDITION DO[2]=ON
2 MOVEL P[1] , 200 mm/s , FINE , SKIP GOTO LABEL[1]
3 MOVEJ P[2] , 12.5% , FINE
4 LABEL[1]
5 MOVEJ P[3] , 12.5% , FINE
```

程序从第 1 行开始执行，当 DO[2]=ON 时，执行到程序第 2 行立刻跳转到第 4 行，从第 4 行继续向下执行；当 DO[2]=OFF 时，程序第 2 行执行结束后从当前行继续向下顺序执行。

3.7 结构指令

结构指令用于控制程序运行流程，包含 CALL 指令、WAIT 指令、WAIT TIME 指令、PAUSE 指令、ABORT 指令、RUN、LOAD、UNLOAD、EXEC 指令。

3.7.1 CALL-调用程序指令

指令格式：Call + 程序名。用以调用子程序，形成嵌套，提高程序的复用性、可读性，减少编程的工作量。如机器人应用过程中，不同位置需要控制末端夹具的动作。

如下图程序初始化与最终位置均调用了 gripperOpen 子程序。

程序名：**CALL**  内容被修改

```
1 CALL gripperOpen
2 MOVEJ P[1: ] 100% FINE
3 MOVEL P[2: ] 4000mm/s FINE
4 CALL gripperClose
5 MOVEC P[3: ] P[4: ] 4000mm/s FINE
6 MOVEJ P[5: ] 100% FINE
7 CALL gripperOpen
8 END PROG
```

图 3.16 Call 指令编程示例

3.7.2 WAIT-等待条件满足指令

指令格式：WAIT+条件+TIMEOUT+SKIP/Waring/被调用的程序

参数说明：

条件可用的数据类型有 IO 类型（DI/DO/UI/UO）与寄存器类型（R[i]/ MI[i]/ MH[i]）

TIMEOUT：

可选参数（非必选参数）。意义是等待时间，单位是秒。可用常数或数值寄存器指定

WAIT 指令不使用附加条件 TIMEOUT 时：执行 WAIT 指令只有当条件满足时，程序才能继续向下执行，否则一直等待直到条件满足为止。

WAIT 指令使用附加条件 TIMROUT 时：执行 WAIT 指令，在指定等待时间内条件满足，程序继续向下执行，在指定等待时间内条件不满足，程序暂停在 WAIT 指令处

SKIP/Waring/被调用的程序：

可选参数（非必选参数）。在使用了可选参数 TIMEOUT 的情况下，可从 SKIP/Waring/被调用的程序 三个参数中选择其中一个可选参数使用

若使用 SKIP 参数：

在在指定等待时间内条件满足，程序继续向下执行，在指定等待时间内条件不满足，跳过当前行（WAIT 指令行），程序继续向下执行。

若使用 Waring：

执行 WAIT 指令，在指定等待时间内条件满足，程序继续向下执行，在指定等待时间内条件不满足，程序暂停在 WAIT 指令处。

若使用被调用的程序，执行 WAIT 指令，在指定等待时间内条件满足，程序继续向下执行，在指定等待时间内条件不满足，执行被调用的程序，被调用的程序执行结束后，程序继续向下执行。

如下图为 wait 指令编程示例。

```
7 | WAIT R[1: default] = 1
8 | WAIT R[1: default] = 1 TIMEOUT=2sec
9 | WAIT R[1: default] = 1 TIMEOUT=1sec SKIP
10 | WAIT R[1: default] = 1 TIMEOUT=1sec Warning
11 | WAIT R[1: default] = 1 TIMEOUT=1sec A
12 | END
```

图 3.17 WAIT 指令编程示例

3.7.3 WAIT TIME-等待时间指令

指令格式：

WAIT TIME + 等待时间，等待时间单位为 sec（秒）。

等待时间可用常数或数值寄存器指定。

等待时间结束之前，机器人停止往下执行指令，直到等待时间结束。等待时间可以是数值或者寄存器。

编程示例如下图所示。

程序名：**WAIT_TIME** *内容被修改*

```
1 R[1: default] = 5
2 MOVEJ P[1: ] 100% FINE
3 CALL gripperClose
4 WAIT 2 sec
5 MOVEJ P[2: ] 100% FINE
6 MOVEL P[3: ] 4000mm/s FINE
7 WAIT R[1] sec
8 CALL gripperOpen
9 MOVEJ P[1: ] 100% FINE
10 END PROG
```

图 3.18 WAIT 指令编程示例

3.7.4 PAUSE-暂停程序指令

指令格式：Pause

当执行到 Pause 指令后，暂停程序的执行，机器人立即规划减速轨迹并进行运动停止，程序状态进入暂停状态。如要继续执行，需要重新按启动键。

如下图所示为 PAUSE 指令编程示例。

程序名：**PAUSE** *内容被修改*

```
1 MOVEJ P[*] 100% FINE
2 MOVEL P[1: ] 4000mm/s FINE
3 MOVEC P[3: ] P[4: ] 4000mm/s FINE
4 IF R[1: default] = 100
5   PAUSE
6 END IF
7 MOVEJ P[5: ] 100% FINE
8 END PROG
```

图 3.19 PAUSE 指令编程示例

3.7.5 ABORT-终止程序指令

格式：ABORT

强制结束指令：结束程序执行，机器人伺服立刻刹车，并上抱闸，伺服母线断电。执行完强制结束指令后，程序进入终止状态，且光标停止在当前行。

如下图所示为 ABORT 指令使用示例。

程序名：**ABORT**  内容被修改


```
1 MOVEJ P[*] 100% FINE
2 MOVEL P[1: ] 4000mm/s FINE
3 MOVEC P[3: ] P[4: ] 4000mm/s FINE
4 # 自定义ERROR CODE = 10001
5 IF R[1: default] = 10001
6   ABORT
7 END IF
8 MOVEJ P[5: ] 100% FINE
9  END PROG
```

图 3.20 ABORT 指令编程示例

3.7.6 RUN-多线程指令

使用 RUN 指令可同时运行多个程序。

格式：RUN + 程序名

程序类型：用户程序（XML 与 JSON）、脚本程序（PYTHON）

编辑指令	编辑工具	快捷操作	关闭
------	------	------	----

RUN	*		
用户程序列表	Basic_Test_IF	Basic_Test_...	Basic_Test_...
用户程序列表	Basic_Test_...	Basic_Test_...	Basic_Test_...
脚本列表	Basic_Test_...	Basic_Test_R	Basic_Test_...
	Basic_Test_...	Basic_Test_...	Basic_Test_...
	Basic_Test_...	Basic_Test_...	Basic_Test_...
	Basic_Test_...	Basic_Test_...	Basic_Test_...
	Basic_Test_...	dome_ta_t...	guest2
	HomeTest	HomeTest_...	init
确认	main	main_chen...	socketwork
取消	softlimit	SpeedTest	t

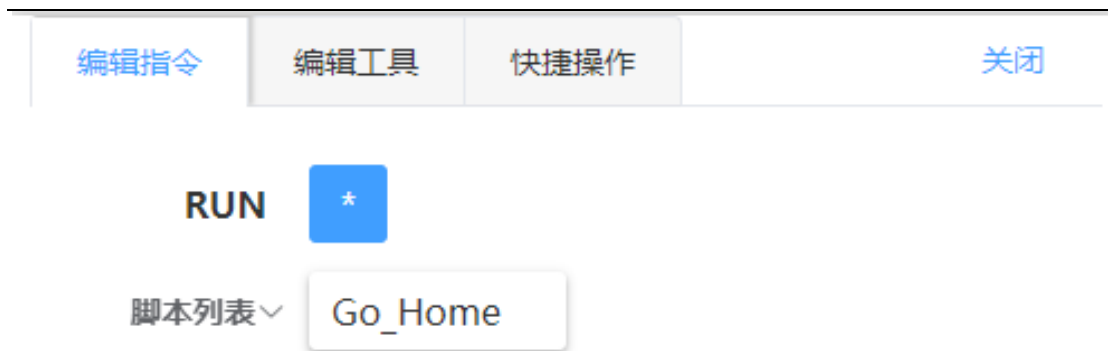


图 3.20 RUN 指令插入示例



说明:

- Main 程序中使用 RUN 指令调用 Pick 程序，则称呼 Main 为“调用方”，Pick 为“被调用方”，且由“被调用方”调用的也为“被调用方”（下同）。
- CALL 指令是同一时间只执行一个程序，RUN 是允许同时执行多个程序，不会等待“被调用方”执行完毕再去执行“调用方”。
- 同一个程序被 RUN 多次，若第一次的执行未结束，则后续的 RUN 不会被响应，会有一个 INFO 事件。
- 仅调用方可写运动指令，被调用方若有运动指令，会报警，报警代码为“program-2328”

恢复按钮 (Reset):

- 针对调用方程序，则被调用方程序也会被恢复。

暂停/终止:

- 如果是调用方，则也会将所有的被调用方暂停或终止。

单步正序:

- 所有程序都是暂停或终止的，才能进行单步
- 在调用方中执行单步正序，在被调用方被执行后，被调用方会跟随调用方一行一行的进行单步正序
- 在被调用方中执行单步正序，则仅有被调用方会一行一行单步正序，调用方不会跟随

单步逆序:

- 忽略 RUN 指令不执行

状态栏:

- 程序名、行号: 有程序在运行, 则显示运行的程序, 否则显示最后运行或者打开的程序
- 机器人运行状态:

如果调用方还在, 则看调用方, 如果不在, 若被调用方有一个在运行, 则为 WORKING, 否则为 ON-STANDBY

UI:

- 启动/恢复: 控制所有程序
- 停止: 控制所有程序
- 终止: 控制所有程序

UO:

- 暂停状态: 同机器人运行状态的描述
- Program Running: 同机器人运行状态的描述

3.7.7 Load-动态程序加载指令

定义: 动态加载程序

格式: Load + 程序 (如“pick”)

在该指令中程序可有寄存器类型 (R[i]/SR[i])、字符串、常数进行指定。



图 3.21 Load 指令插入示例

3.7.8 Exec-执行动态程序加载指令

定义：执行程序

格式：Exec + 程序（如“pick”）

该指令用于执行 Load 动态程序加载的程序。在该指令中程序名可由寄存器类型（R[i]/SR[i]）、字符串、常数进行指定。



图 3.22 Exec 指令插入示例

3.7.9 Unload--动态程序卸载指令

定义：卸载程序

格式：Unload + 程序（如“pick”）

在该指令中程序可有寄存器类型（R[i]/SR[i]）、字符串、常数进行指定。



图 3.23 Unload 指令插入示例

3.8 其他指令

3.8.1 TF_NO-工具坐标系指令

用于程序中切换工具坐标系，一般与 UF_NO.配合使用。

指令格式：TF_No = 工具坐标系号；“工具坐标系号”数据类型有寄存器类型（R）和常数两种可以选择



由于 PR 寄存器不带 UF/TF 信息，程序执行时表示指定 TF 在当前激活使用 UF 下的位姿信息；所以切换不同的 UF/TF，实际机器人在空间中位置可能不同。

P 表示局部位姿数据，包含 UF/TF 信息，如果运动指令使用 P 数据时，UF_No.、TF_No.与 P 数据记录不一致，则程序无法继续向下执行，需要在该运动指令前修改 UF_No.、TF_No.使其与 P 数据记录一致。

3.8.2 UF_NO 用户坐标系指令

用于程序中切换用户坐标系，一般与 TF_NO 配合使用。

指令格式：TF_No = 用户坐标系号；“用户坐标系号”数据类型有寄存器类型（R）和常数两种可以选择



由于 PR 寄存器不带 UF/TF 信息，程序执行时表示指定 TF 在当前激活使用 UF 下的位姿信息；所以切换不同的 UF/TF，实际机器人在空间中位置可能不同。

P 表示局部位姿数据，包含 UF/TF 信息，如果运动指令使用 P 数据时，UF_No、TF_No 与 P 数据记录不一致，则程序无法继续向下执行，需要在该运动指令前修改 UF_No、TF_No 使其与 P 数据记录一致。

3.8.3 J_POS 当前关节坐标指令

关节坐标系中的当前位置，常与位姿寄存器 PR 搭配使用

例

PR[i] = J_POS 将机器人关节坐标系中的当前位置赋值给位置寄存器 PR[i]

3.8.4 L_POS 当前笛卡尔坐标指令

直角坐标系中的当前位置，常与位姿寄存器 PR 搭配使用

例

PR[i] = L_POS 将机器人关节坐标系中的当前位置赋值给位置寄存器 PR[i]

3.8.5 Payload_NO-负载设定指令

用于在程序中切换负载编号，如应用过程中，抓取工件前，机器人手臂末端只有夹具本身的负载；抓取工件后，机器人手臂末端负载是夹具及工件的综合；将抓取前后的负载数据记录在以不同的负载编号记录。

指令格式：PAYLOAD_NO = 负载记录编号；“负载记录编号”数据类型有寄存器类型（R）和常数两种可以选择

3.8.6 Timer-计时器指令

程序计时器指令：用于启动或停止程序计时器。可以通过计时器的状态来控制计

时器的启停。全局计时器支持多程序（任务）共享。

指令格式

1. `Timer[i]=值`。值的数据类型有四种，分别是状态（Start/Stop/Reset）、寄存器（R）、常数、方法（请参考 3.10 节）

当值是状态时

`Timer[i]=Start`; 计时器开始计时

`Timer[i]=Stop`; 计时器停止计时

`Timer[i]=Reset`; 计时器复位（数值清零）

[i]中的 i 表示计时器号，可以用常数直接指定，或者用数值寄存器 R 间接指定

2. 赋值：`R[i]=Timer [i]`（前提条件是 Timer 必须被定义过状态，如未定义则 Timer 为空值无法赋值）。

3.8.7 Comment-注释指令

注释指令用来在程序里加入注释，提高程序的可读性，该注释对于程序运行没有任何影响

格式：`#`（注释文本）

注释文本支持所有数据类型

3.8.8 OVC-全局速度指令

用以修改全局速度的指令

指令格式：`Overall_Velocity_Coefficient =速度百分比`。“速度百分比”的数据类型有数值寄存器类型（R）和常数两种可以选择

3.8.9 OAC-全局加速度指令

用以修改全局加速度的指令

格式：`Overall_Acceleration_Coefficient =加速度百分比`。

参数：

“加速度百分比”的数据类型有数值寄存器类型（R）和常数两种可以选择。范围 10%至 100%（修改此数值仅可由厂商权限进行！）

例子：

`Overall_Acceleration_Coefficient = 80%`

Overall_Acceleration_Coefficient = R[1]

3.8.10 LABEL-标签指令

定义程序标签，常与 GOTO 指令（见 3.6.4 节）和 SKIP CONDITION 指令（见 3.6.5 节）搭配使用。

格式：LABEL[i : Comment]; 其中 i 是标签号，Comment 是标签注释信息

3.8.11 Socket-套接字指令

指令概述：

SocketOpen: 使用 SocketOpen 指令创建一个 Server 并等待 Client 连接

SOCKET OPEN SK[*], parameter

参数 SK[*]表示当前所用的套接字设备，可选参数 parameter 表示语句执行结果（见第 10 章节 Socket 错误代码列表），以数值寄存器(R[*])表示。

补充说明：

- SocketOpen 为其他 socket 命令使用前提条件，请先调用 SocketOpen 指令后再进行后续 socket 操作。

SocketConnect: 使用 SocketConnect 指令创建一个 Client，并连接一个 Server

SOCKET CONNECT SK[*], parameter

参数 SK[*]表示当前所用的套接字设备，可选参数 parameter 表示语句执行结果（见 Socket 错误代码列表），以数值寄存器(R[*])表示

SocketRecv: SocketRecv 指令用于从 Socket 读取字符，可以指定最大长度
SOCKET RECV SK[*], R[*], SR[*], parameter2

参数 SK[*]表示当前所用的套接字设备，参数 R[*]表示接受长度；SR[*]表示接收到的数据存在字符串寄存器(SR[*])中，可选参数 parameter2 表示语句执行结果（见 Socket 错误代码列表），以数值寄存器(R[*])表示，

补充说明：

- Server 或 Client 都可调用该指令
- 接收字符串的长度：最大值为 254
- 指令中的 SK 寄存器中配置的“接受默认超时时间”参数，对该指令有效

SocketSend: SocketSend 指令用于写入字符串到 Socket，从而发送给对端

SOCKET SEND SK[*], "", parameter2

参数 SK[*]表示当前所用的套接字设备，参数 ""表示需要发送的内容(字符串)，可以是常量，也可以是字符串寄存器(SR[*])，可选参数 parameter2 表示语句执行结果(见 Socket 错误代码列表)，以数值寄存器(R[*])表示，

补充说明：

- Server 或 Client 都可调用该指令
- 若待发送的字符串使用 SR 表示，则最大长度为 254
- 可以直接发送常量字符串，也可以将字符串存入到字符串寄存器中，通过选择该寄存器的方式发送内容

SocketClose: SocketClose 指令用于关闭连接；
SOCKET CLOSE SK[*], parameter

参数 SK[*]表示当前所用的套接字设备，可选参数 parameter 表示语句执行结果(见 Socket 错误代码列表)，以数值寄存器(R[*])表示



可选参数用也可以，不用也可以，根据实际需求选择是否使用，可选参数以外的参数，是必须要有的。

3.8.12 复合运算指令

复合运算指令，可以将 TP 程序的赋值语句、条件比较语句、以及等待指令语句中各类算符和数据组合起来。复合运算指令支持括弧“ () ”。

复合运算指令，可以在寄存器指令、IF 指令、WAIT 指令、WHILE 指令中使用。

复合运算指令，如下所示那样指定在括弧内。

- IF DI[1] = (DO[2] AND DO[3])

执行语句

END IF

句子中没有括弧时，成为通常的运算指令。

- WAIT (DO[1] = ON AND D0[2] = OFF) OR (R[1] = 1 OR R[2] = 2)

数据类型

复合运算指令，可以使用如下数据类型。

类型	值	数据

数值	可作为数据处理数值。整数和实数，两者都可以使用。	寄存器、常数、位置寄存器的要素
布尔	数据可设定为 ON 或 OFF 的任一值	DI/O、UI/O、ON、OFF

算符

复合运算指令，可以使用如下算符。

算符	运算
+	左边和右边的加法运算
-	左边和右边的减法运算
*	左边和右边的乘法运算
/	左边和右边的除法运算
MOD	左边和右边的除法运算余数
DIV	左边和右边的除法运算的商的整数部分

- 算术算符只可以使用数值类型的数据。
- 算术算符的输出数据，始终为数值型。

算符	运算
AND 左边和右边的逻辑积	左边和右边的逻辑积
OR 左边和右边的逻辑和	左边和右边的逻辑和

- 逻辑算符只使用于布尔型数据。
- 逻辑算符的输出数据始终为布尔型。

算符	运算
=	左边和右边相等时，返回 ON。左边和右边不相等时，返回 OFF。
<>	左边和右边不相等时，返回 ON。左边和右边相等时，返回 OFF。
<	左边小于右边时，返回 ON。左边大于右边时，返回 OFF。
>	左边大于右边时，返回 ON。左边小于右边时，返回 OFF。
<=	左边小于右边或者等于右边时，返回 ON。左边大于右边时，返送 OFF。
>=	左边大于右边或者等于右边时，返回 ON。左边小于右边时，返送 OFF。

- “=”和“<>”，在数值型数据和布尔型数据中都可以使用。
- “<”、“>”、“<=”、以及“>=”，只可在数值数据中使用。

下表示出算符的优先顺序。

优先顺序	算符
高	*、/、DIV、MOD
	+、—
中	<, >, <=, >=
	=, <>
	AND
低	OR

条件语句

下面示出条件指令中的复合运算指令的使用例。

IF (R[1] = (PR[1, 6] + R[1]) * R[2])

IF (DI[1] AND (DI[2] OR DI[3]))

- 可以在条件语句中使用复合运算式。
- 条件语句的结果必须是布尔型。

等待指令

下面示出等待指令中的复合运算指令的使用例。

WAIT (DI[1] = ON AND (DI[2] = ON OR DI[3] = ON))

- 可以在等待指令的条件语句中指定复合运算式。
- 条件语句的结果必须是布尔型。
- 等待指令，在表达式结果成为 ON 之前等待。

复合运算指令的追加和修改

具有复合运算功能的指令，指令编辑界面符号说明，并使用下表符号进行复合运算指令的追加和修改

复合运算指令的追加和修改使用符号表

	代表增添一个表达式
	代表去除一个表达式（至少要有有一个，即等式右边至少要有一项）
	代表增加括号
	代表去除括号

3.9 字符串指令

3.9.1 StrLen-字符串长度获取指令

用于取得字符串的长度，并将结果带入寄存器中，常用方式为 $R[i] = \text{StrLen}$ (值)。

值的数据类型通常为常量字符串或字符串寄存器 (SR)。

举例如下：

$R[i] = \text{StrLen}$ ("666")，则 $R[i]$ 的值为 3

$R[i] = \text{StrLen}$ (SR[i])，其中 SR[i] = "111111"，则 $R[i]$ 的值为 6

3.9.2 FindStr-字符串搜索指令

$R[i] = \text{FindStr}$ (值 1, 值 2)；值 1 与值 2 的数据类型通常为常量字符串或字符串寄存器 (SR)。

值 1 表示“对象字符串”，值 2 表示“需要搜索的字符串”， $R[i]$ 表示需要搜索的字符串在对象字符串中的位置。从左向右，对象字符串中的第一个字符的位置是 0，以此类推。没有找到需要搜索的字符串时， $R[i]$ 的值为 -1。

举例如下：

$R[i] = \text{FindStr}$ ("123456", "1") $R[i]$ 的值为 0

$R[i] = \text{FindStr}$ ("123456", "0") $R[i]$ 的值为 -1

3.9.3 SubStr-字符串截取指令

$SR[i] = \text{SubStr}$ (值 1, 值 2, 值 3)。

值 1 表示“对象字符串”，值 2 表示“起始点位置”，值 3 表示“需要截取的字符串的长度”。其中值 1 的数据类型通常为常量字符串或字符串寄存器 (SR)，值 2 与值 3 的数据类型通常为常数或数值寄存器 (R)。SR[i] 表示截取后的字符串。

$SR[i] = \text{SubStr}$ (值 1, 值 2, 值 3) 指令，表示从对象字符串中取得部分字符串，将其结果代入字符串寄存器。截取后的字符串，根据从对象值的第几个字符这样的起始点位置、以及需要截取的字符串的长度来决定。

举例如下：

$SR[i] = \text{SubStr}$ ("P123456", 0, 1) $SR[i]$ 的值为 "P"

3.10 方法（函数）

3.10.1 SIN-正弦函数

Sin (Sine) 用于计算一个角值的正弦值。

基本示例

以下示例介绍了函数 Sin。

```
R[1:]=Sin (*)
```

R[1:]将获得*的正弦值。正弦值的范围= (-1, 1)。

3.10.2 COS-余弦函数

Cos (Cosine) 用于根据有关数据类型 num 的角度值，计算余弦值。

基本示例

以下示例介绍了函数 Cos。

```
R[1:]=Cos (*)
```

R[1:]将获得*的余弦值。余弦值的范围= (-1, 1)。

3.10.3 TAN-正切函数

Tan (Tangent) 用于计算一个角值的正切值。

基本示例

以下示例介绍了函数 Tan。

```
R[1:]=Tan (*)
```

R[1:]将获得*的正切值。

3.10.4 ARCSIN-反正弦函数

ArcSin 用于计算一个角值的反正弦值。

基本示例

以下示例介绍了函数 ArcSin。

```
R[1:]= ArcSin(0.5)
```

R[1:]将获得 0.5 的反正弦值，为 30。

3.10.5 ARCCOS-反余弦函数

ArcCos 用于计算一个角值的反余弦值。

基本示例

以下示例介绍了函数 ArcCos。

```
R[1:]= ArcCos (-0.5)
```

R[1:]将获得-0.5 的反余弦值，为 120。

3.10.6 ARCTAN-反正切函数

ArcTan 用于计算一个角值的反正切值。

基本示例

以下示例介绍了函数 ArcTan。

```
R[1:]= ArcTan(1)
```

R[1:]将获得 1 的反正切值，为 45。

3.10.7 ABS-绝对值函数

Abs 用于获取绝对值，即数字数据的正值。

基本示例

以下示例介绍了函数 Abs。

```
R[1:]=Abs (R[2:])
```

将 R[1:]指定为 R[2:]的绝对值。

3.11 碰撞检测指令

3.11.1 CollisionDetect

用于设置碰撞检测开关

指令格式：CollisionDetect ON/OFF , GROUP * , AXIS*；其中 GROUP 与 AXIS 参数是可选参数。

可选参数 GROUP *：运动组，机器人本体运动组参数是 1（目前不支持外部轴配置）。

可选参数 **AXIS ***: 轴, 在运动组是 1 的情况下表示机器人本体的轴 1-6 轴 (根据实际情况选择轴)。

不使用可选参数的情况下表示开启/关闭所有运动组的碰撞检测。

基本示例

以下示例介绍了 **CollisionDetect** 指令。

CollisionDetect ON , GROUP 1 , AXIS 1; 表示打开机器人本体 1 轴的碰撞检测。

CollisionDetect ON; 表示打开机器人本体所有轴的碰撞检测。

3.11.2 CollisionRange

用于设置碰撞检测偏差值。碰撞检测偏差值越大, 机器人碰撞检测越灵敏; 反之, 越不灵敏。

指令格式: **CollisionRange *% , GROUP * , AXIS***; 其中 **GROUP** 与 **AXIS** 参数是可选参数。

可选参数 **GROUP ***: 运动组, 机器人本体运动组参数是 1 (目前不支持外部轴配置)。

可选参数 **AXIS ***: 轴, 在运动组是 1 的情况下表示机器人本体的轴 1-6 轴 (根据实际情况选择轴)。

不使用可选参数的情况下表示所有运动组的碰撞检测灵敏度为*%。

基本示例

以下示例介绍了 **CollisionRange** 指令。

CollisionRanget 30% , GROUP 1 , AXIS 1; 表示机器人本体 1 轴的碰撞检测灵敏度为最大灵敏度的 30%

CollisionRanget 30%; 表示机器人本体所有轴的碰撞检测灵敏度为最大灵敏度的 30%

3.12 码垛指令

码垛指令是一套指令及其附带信息的集合的统称, 包含:

指令	描述
码垛格式指令	根据码垛寄存器的值和码垛模式计算当前码垛点的位置，根据路径条件匹配对应路径，重写码垛动作指令的位置数据。
码垛动作指令	使用接近点/取放点/退出点作为位置数据的动作指令。
码垛结束指令	计算下一个堆栈点，重写码垛寄存器的值。

3 条指令必须同时存在同一个程序（主程序）中才能起作用，将其中任何一个指令放置在 CALL 或 RUN 的程序中，均忽略不可执行。有关 PL 寄存器的赋值操作，要在码垛格式指令前完成，否则可能会发生机器人与垛碰撞或松手爪时的工件位置过高，导致工件损坏的情况。码垛格式指令和节数指令之间可任意插入其他指令，不影响码垛功能的执行。

3.12.1 码垛格式指令

码垛格式指令和结束指令均用于标识码垛指令的作用区间，二者成对存在。码垛格式指令中仅显示描述垛型特征的参数，便于阅读，不可修改。

指令形式：

PALLETIZING[i: NAME], B,LOAD, R, C, L

指令含义

参数	描述
B	垛形态类型有 B、E;
i	码垛指令 ID，范围[1,30]整数;
NAME	码垛名称，继承“菜单-应用-码垛”中的具体码垛工艺工程的名称;
LOAD	码垛、拆垛标识，值为“LOAD”：码垛，“UNLOAD”：拆垛;
R, C, L	行列层数值， $N \in [1,255]$;

举例：

PALLETIZING [1: Test], B, UNLOAD, 3, 4, 5

代表使用一个 3 行 4 列 5 层的 B 型垛，ID 为 1，名称为 Test，模式为 B 型拆垛的码垛工艺工程。

3.12.2 进入点/取放点/退出点动作指令

用于记录示教得到的点位数据，这些点位将根据示教的初始码放位姿进行偏移，偏移向量由 RCL 自加或自减规则以及生成的码放点位决定。在多条路径样式情况下，[R,C,L]点位的路径样式使用满足匹配路径条件的路径样式。码垛动作指令仅支持 MOVEJ（默认），MOVEL。

进入点，取放点和退出点指令不可脱离码垛格式指令和码垛结束指令的作用范围独立存在。

指令形式：

PAL [i, A_x]

PAL [i, BTM]

PAL [i, R_x]

指令含义

参数	描述
PAL [...]	码垛专用位姿寄存器（效果与 P 类似）
i	与具体码垛工艺工程绑定，范围 $N \in [1,30]$
A_x:	进入点 ID，A 为固定前缀，x 为正整数，点位数与对应的工艺工程中设置的接近点个数相同；
BTM:	码放点，每个路径点上只有一个
R_x:	退出点 ID，R 为固定前缀，x 为正整数，点位数与对应的工艺工程中设置的退出点个数相同；

举例：

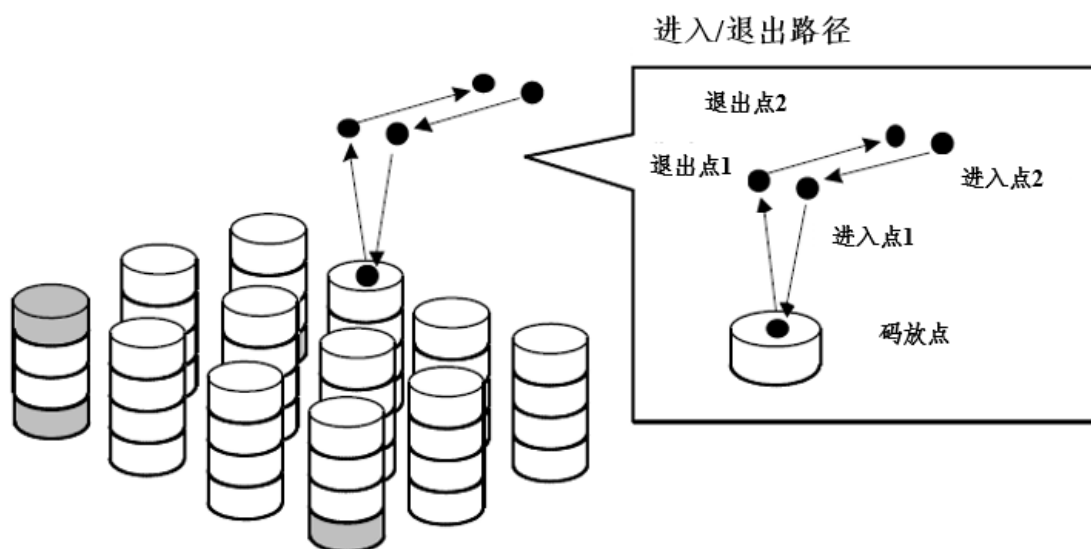
MOVEJ PAL[1, A_2], 90%, SD

MOVEJ PAL[1, A_1], 500mm/s, SD

MOVEJ PAL[1, BTM], 500mm/s, FINE

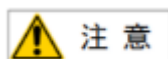
MOVEJ PAL[1, R_1], 500mm/s, FINE

MOVEJ PAL[1, R_2], 500mm/s, FINE



进入点，码放点和退出点构成的路径

表示先运动到无障碍进入码放位置，再运动到码放点位，到达后可释放工件，最后运动到无障碍退出点位，从该点机器人可安全返回。



实际使用过程中，可以根据实际情况跳点，比如 2 接近点 3 退出点的某路径，可以使用[0,2]个接近点和[0,3]的退出点，例如：

MOVEJ PAL[1, A_1], 500mm/s, SD

MOVEJ PAL[1, BTM], 500mm/s, FINE

MOVEJ PAL[1, R_1], 500mm/s, SD

MOVEJ PAL[1, R_3], 500mm/s, FINE

3.12.3 码垛结束指令

用于标识码垛指令的作用区间，与码垛格式指令成对存在。同时，程序运行到本指令时，将按照码垛格式指令中配置的自加（自减）规则，对码垛寄存器值进行更新。该指令的属性从属于码垛格式指令，不可单独配置。

指令形式：

`PALLETIZING_END [i: NAME]`

指令含义

参数	描述
i:	码垛指令 ID，配置码垛格式指令的时候自动设定，不可单独更改。
NAME:	码垛指令名称，继承“菜单-应用-码垛”中的具体码垛工艺工程的名称；

举例：

```

PL[1] = [2,2,2]
PALLETIZING [1: Test ], B, LOAD, 3, 4, 5
MOVEJ PAL[1, A_1], 90%, SD
MOVEL PAL[1, BTM], 500mm/s, FINE
MOVEL PAL[1, R_1], 500mm/s, FINE
PALLETIZING_END [1: Test]
    
```

表示码垛顺序行->列->层,正序，步进值 1。1-5 行，PL[1] 的值为[2,2,2]；第 6 行结束时，PL[1] 的值为[3,2,2]

注意：

同一个程序中，可使用多个码垛工艺工程，并且根据程序进行嵌套使用，例如：

```

PALLETIZING [1: Test ], B, LOAD, 2, 10, 5
MOVEJ PAL[1, A_1], 90%, SD
MOVEL PAL[1, BTM], 500mm/s, FINE
MOVEL PAL[1, R_1], 500mm/s, FINE
.....
PALLETIZING [2: Test ], B, LOAD, 3, 10, 5
    
```

```

MOVEJ PAL[2, A_2], 90%, SD
MOVEJ PAL[2, A_1], 90%, SD
MOVEL PAL[2, BTM], 500mm/s, FINE
MOVEL PAL[2, R_1], 500mm/s, FINE
MOVEL PAL[2, R_2], 500mm/s, FINE
.....
PALLETIZING_END [1: Test]
PALLETIZING_END [2: Test]
    
```

3.13 Vision 指令

TP 端的程序使用视觉指令，需要修改视觉控制器的 IP 地址，保证视觉控制器 (PC) 的 IP 地址与 TP 端 (机器人控制柜) 的 IP 地址处于同一个局域网网段中。

TP 端 (机器人控制柜)	视觉控制器 (PC)	
IP 地址	192.168.110.2	192.168.110.3
子网掩码	255.255.255.0	255.255.255.0

3.13.1 VISION RUN_FIND 启动视觉程序指令

用于启动一个视觉程序。

软触发指令格式: VISION RUN_FIND + '视觉程序名称'

硬触发指令格式: VISION RUN_FIND

3.13.2 VISION GET_QUANTITY 获取视觉检出个数指令

用于获取视觉检出个数。

指令格式: VISION GET_QUANTITY + '视觉程序名称' + R[i]

例: VISION GET_QUANTITY test, R[1:] 视觉程序 test 检测出个数赋值给数值寄存器 R[1]

3.13.3 MODELID 获取模板 ID 指令

该指令将视觉寄存器的模板 ID 赋值给数值寄存器 R[i]。

指令格式：R[i] = VR[i, MODELID]

例：R[1]=VR[1, MODELID]

3.13.4 GET_OFFSET 获取检出位置指令

用来从视觉程序中获取检出结果，将其存储至指定视觉寄存器中。

指令格式：VISION GET_OFFSET + '视觉程序名称' + VR[i] +GOTO LABEL[i]

说明：当视觉程序还未处理完，则等待到图像处理。

当视觉程序未检出，则根据 GOTO，跳转到程序指定标签。

当视觉程序检出多个结果时，可反复执行该指令。

例：VISION GET_OFFSET test, VR[1], GOTO LABEL [1]

3.13.5 VOFFSET 视觉位置补偿指令

该指令用存储在视觉寄存器中的数据补偿机器人的位置

指令格式：..., VOFFSET,VR[i]

例：MOVE PR[1:], 500mm/s, FINE, VOFFSET VR[1]

3.13.6 检出的位置信息赋值指令

将检出视觉寄存器 VR[i]的位置信息赋值给位姿寄存器 PR[i]

指令格式：PR[i] = VR[i]

3.13.7 视觉程序举例

例 1 一次拍照检出多个同样的工件

Label[1]

VISION RUN_FIND test

VISION GET_QUANTITY test, R[1:]

IF R[1]=0, GOTO LABEL [1]

While R[1]>0,

VISION GET_OFFSET test, VR[1], GOTO LABEL [1]

MOVE PR[1:], 500mm/s, FINE, VOFFSET VR[1]

R[1]=R[1]-1

End

例 2 拍照检出不同工件

Label[1]

VISION RUN_FIND test

VISION GET_OFFSET test, VR[1], GOTO LABEL [1]

R[1]=VR[1, MODELID]

Switch R[1]:

Case 1:

 MOVEL PR[1], 500mm/s, FINE, VOFFSET VR[1]

Case 2:

 MOVEL PR[2], 500mm/s, FINE, VOFFSET VR[1]

End

4. 程序的创建与执行

程序的创建、修改、删除等操作必须在手动模式下进行，自动模式下不能操作！

4.1 创建程序

创建程序有两种方式：

1. 依次点击“菜单按钮”→“程序”进入界面如图 4.1，点击“新增程序”，弹出如图 4.2 所示新增程序界面；在该界面填入程序名称、程序备注、选择程序类型后，点击“确认”即可完成程序新建。

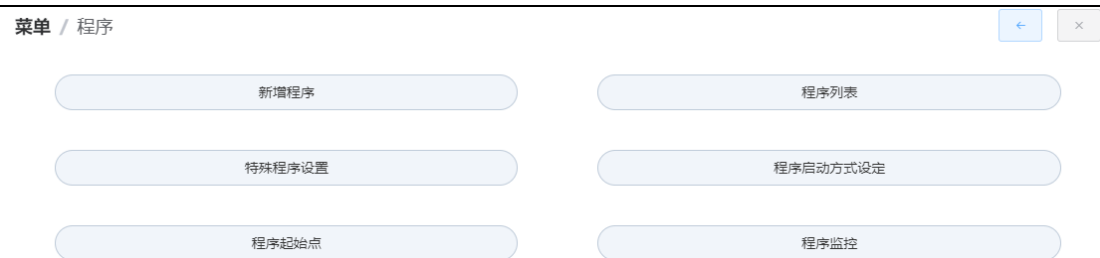


图 4.1 程序菜单界面



图 4.2 新建程序窗口

2. 第二种，依次点击“菜单按钮”→“程序”→“程序列表”进入图 4.3 所示界面，点击“新增程序”，同样出现如图 4.2 在该界面填入程序名称、程序备注、选择程序类型后，点击“确认”即可完成程序新建。

#	程序名称	程序类型	修改时间	程序作者	程序备注	程序保护
1	A					<input type="checkbox"/>
2	aa	None	2023-02-09 14:34:12	admin		<input type="checkbox"/>
3	AAA	None	2023-02-09 19:30:55	admin		<input type="checkbox"/>

图 4.3 程序列表界面

4.1.1 程序操作界面说明

打开新建的程序如图 4.4 所示，只有空白行及“END”行。



图 4.4 程序编辑界面

界面按键说明：

“返回列表”	返回程序列表界面
“编辑指令”	<p>表示对已经存在程序指令可以进行编辑：</p> <p>上移/下移：将当前程序行指令移至上一行之前或者下一行之后；</p> <p>复制/剪切：将当前程序行指令复制或者剪切到剪贴板；</p> <p>粘贴：将剪切板的内容粘贴到当前程序行的下一行；</p> <p>撤销/重做：回退或者恢复撤销重复上一次的操作；</p> <p>禁用/启用：对当前程序行指令进行禁用或者启用，也可方便用于调试；</p> <p>添加空白行：在当前程序行的下一行新添加一行空白行。</p>
“插入指令”	表示插入一条新的指令，例如“常用指令、运动指令、逻辑指令、赋值指令、IO 指令、结构指令、其他指令”。
“箭头指向光标”	<p>表示将程序的初始运行程序行号移动到光标指向的程序行号，</p> <p>例如：下图中程序运行的初始程序行号不在是第一行，而是第 5 行。执行光标是个箭头，在程序行号最左边，指示着程序行号。</p> <p>只能在手动模式下才能进行此操作。</p>

	<p>程序名 : aaa</p> <pre> 1 2 MOVEJ PR[1: default] 500mm/s FINE 3 4 CALL TEST_MLPCS 5 WAIT R[1] sec 6 7 MOVEJ P[1: first] 12.5% FINE 8 MOVEJ P[1: first] MR[1: default]% SD4, ACC=9, OFFSET=PR[2] UF[3], TOOL_OFFSET=PR[2] TF[2] 9 WAIT DI[*] = ON TIMEOUT=R[2]sec TEST_MLPCS 10 END CONDITION OFFSET 11 SWITCH UO[*] 12 CASE * 13 IF * * * * 14 15 END IF 16 TIMER[*] = <exp> 17 18 IF * = * 19 20 ELIF * = * 21 22 END IF 23 END SWITCH 24 MOVEJ PR[MR[1: default]] 500mm/s FINE 25 MOVEJ P[*] 500mm/s FINE 26 END PROG </pre> <div style="float: right; border: 1px solid black; padding: 2px; margin-top: 10px;"> 点位列表 IO 状态 寄存器 </div>
“跳转到”	表示输入程序中对应的程序行号，点击“go”将光标移动到该程序行号。
“连续执行”	默认选择，表示按照正序方向、连续地执行程序，直到将被选择程序执行完成或者被暂停/终止。
“单步正序”	表示单步执行，正序地、一条指令一条指令地执行程序，每按一下运行键执行一条指令，执行完以后程序执行状态处于“暂停”。 只能在手动模式下才能进行此操作。
“单步逆序”	表示逆序执行，逆序地、一条指令一条指令地执行程序，每按一下运行键执行一条指令，执行完以后程序执行状态处于“暂停”。 逆序执行时，只会执行运动指令，不能执行控制指令（除运动指令以外的指令） 只能在手动模式下才能进行此操作。
“点位列表”	通常情况下点位列表隐藏在程序编辑界面的右边，如图 4.5 所示；只有当用户点击点位列表才出现。点位列表包含该程序的私有 P 点和全局 PR 点。这些点在编程时被更改或在 PR 寄存器界面被修改后，点位列表界面都应该随之改变；反之亦然。此外，在伺服上电状态下，在点位列表界面选中一个点位后，可以长按左下角的“移动”按钮，将机器人缓慢移动到该位姿。
“IO 状态”	正常情况下该界面隐藏在编程界面右边，如图 4.5 所示，当用户点击 IO 状态按钮后弹出。此界面内容与“通讯→IO 状态”界面内容一

	致且数据共享。目的是方便用户在编程调试过程中对 IO 状态进行监控。
“寄存器”	正常情况下该界面隐藏在编程界面右边，如图 4.5 所示，当用户点击寄存器按钮后弹出。此界面内容与寄存器的相关界面内容一致且数据共享。目的是方便用户在编程调试过程中对寄存器状态进行监控。

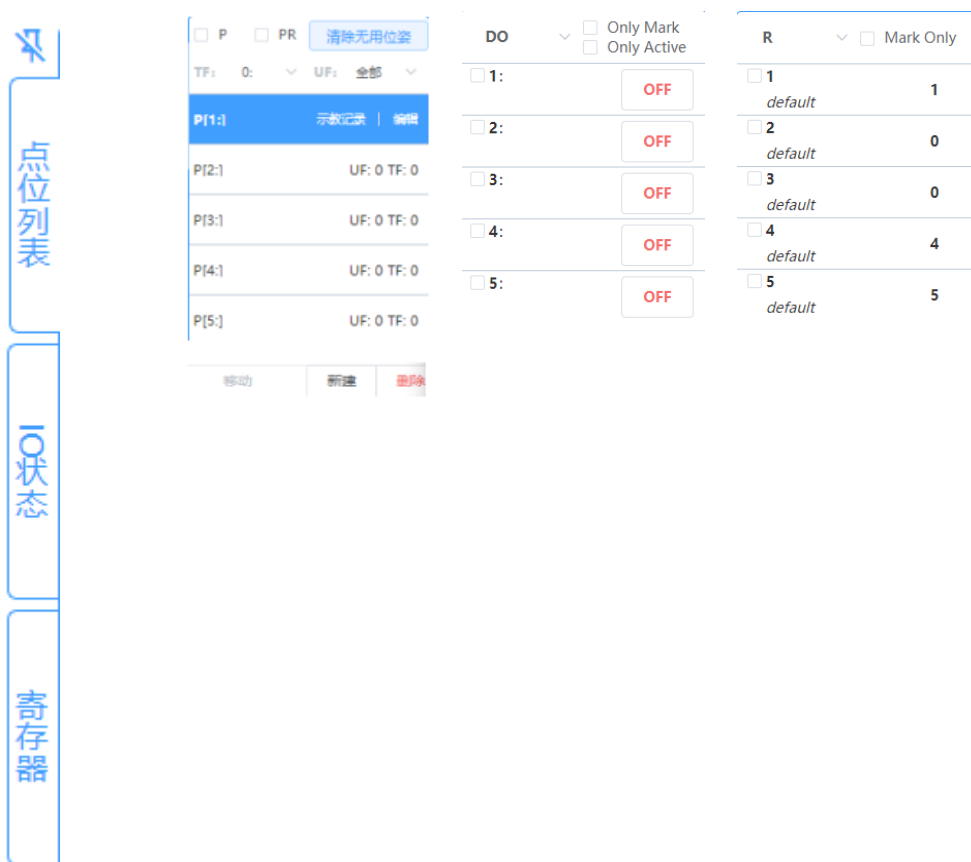


图 4.5 从左往右分别是点位列表、IO 状态列表、寄存器列表

点位列表、IO 状态、寄存器界面功能键说明：

功能键	具体功能
	点击此处可以把“点位列表”、“IO 状态”、“寄存器”界面锁定在程序编辑画面，在点击即可解锁。方便调试时监视相关数据状态

	在点位列表画面左上方点击此处，可以只显示 P 变量点位
	在点位列表画面正上方点击此处，可以只显示 PR 变量点位
	在点位列表画面右上方点击此处，可以删除想删除的 P 和 PR 点位，也可一键清除所有点位
	在点位列表画面点击此处，可以选择查看以哪个工具坐标系为基准建立的点位
	在点位列表画面点击此处，可以选择查看以哪个用户坐标系为基准建立的点位
	在点位列表画面选中想操作的点位，此处可以重新示教点位数据和手动编辑点位数据
“移动”	机器人手动模式伺服上电的前提下，在点位列表画面选择需要移动的点位，点击此处，可以移动机器人到目标点位
“新建”	在点位列表画面点击此处，可以新建机器人点位数据
“删除”	在点位列表画面点击此处，可以删除想要删除的点位数据
	在 IO 状态画面点击此处，可以切换需要监视的 IO 类型
	在 IO 状态画面点击此处，可以屏蔽所监视的 IO 状态信息
	在 IO 状态画面点击此处，可以屏蔽没有配置的 IO 状态，只显示已经配置成功的 IO 状态
	在寄存器画面点击此处，可以切换到 MR、SR 寄存器监视画面
	在寄存器画面点击此处，可以屏蔽所监视的寄存器状态

4.1.2 复制程序

在程序列表界面选中需要复制的程序，点击“复制程序”，即如图 4.6 所示的画面，用户可以复制当前选定的程序并重新定义复制程序的程序名称、程序备注和程序类型。（注：程序名不能与当前程序列表中的程序名称一致），复制程序只对程序员以上级别开放。



图 4.6 复制程序窗口

4.1.3 删除程序

在程序列表界面选中需要删除的程序，点击“删除程序”，出现如图 4.7 所示的界面，如点击确定则该程序将被永久删除，无法恢复。删除程序只对程序员以上级别开放。



图 4.7 删除程序窗口

4.1.4 选择程序

依次点击“菜单按钮”→“程序”进入界面如图 4.8 所示，点击“程序列表”，进入如图 4.9 所示程序列表界面，点击想要选择的程序如图 4.10 所示，再点击“打开程序”进入当前程序编辑界面如图 4.11 所示。

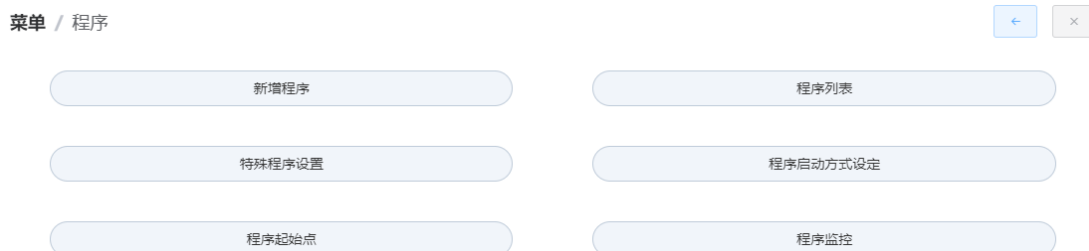


图 4.8 程序菜单窗口

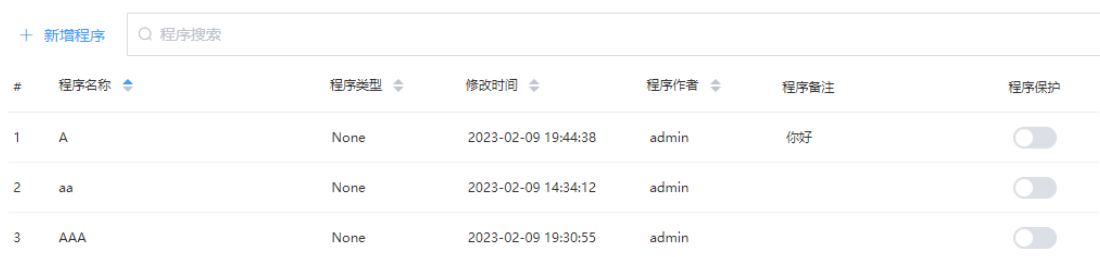


图 4.9 程序列表窗口

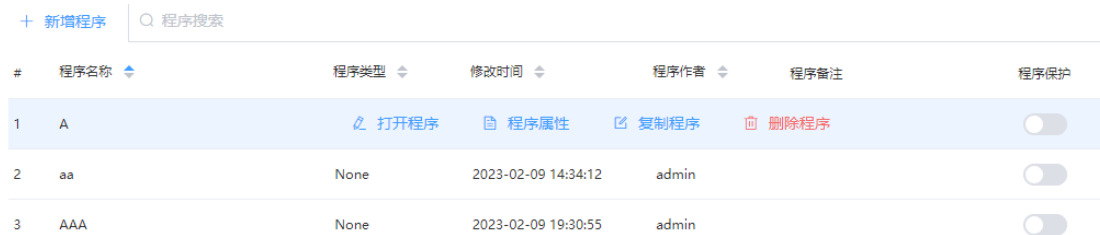


图 4.10 选中程序



图 4.11 程序编辑窗口

4.1.5 创建动作指令

插入动作指令步骤如下：

1. 在程序操作界面点击选中需要插入程序位置的程序行，选中程序行后。程序行数字前会有光标显示，如图 4.12 所示。

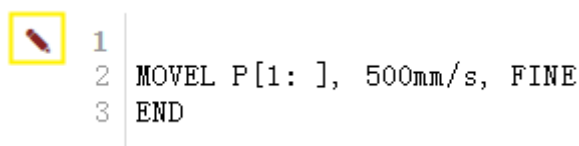


图 4.12 笔型光标

2. 点击程序操作界面的“插入指令”进入指令选择界面，如图 4.13 所示→选择需要的运动指令（此处以 Move Line 为例），进入运动指令编辑界面。



图 4.13 指令选择窗口

3. 根据实际情况修改运动指令内容。




图 4.14 运动指令编辑窗口

4. 把机器人示教到希望的目标位置，点击图 4.14 中的新建位姿后 P[*]中的“*”将会显示具体的数字，如图 4.15 所示，点击确认，即可完成运动指令的插入（MOVEJ、MOVEC 指令的插入步骤相同）。



图 4.15 完整运动指令示例

补充说明：

5.在运动指令编辑界面点击  可以切换运动指令 (MOVJ、MOVEC)，若想对 P 点进行编辑，在步骤图 4.14 中点击“P[*]”显示画面如下

MOVE LINE 



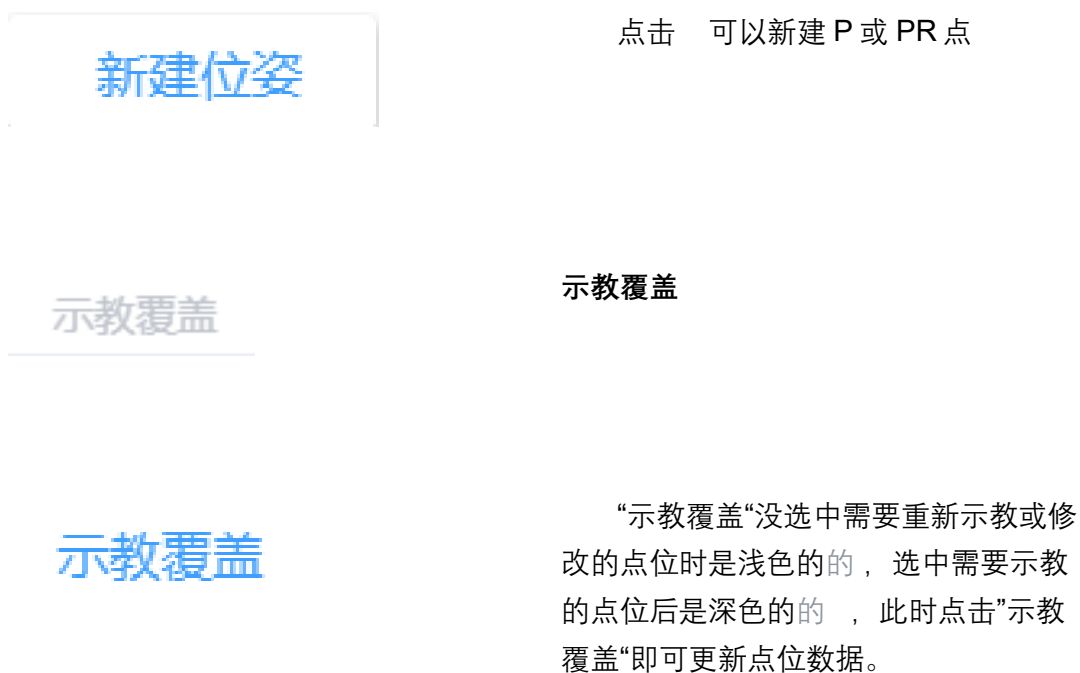
图 4.16 新建点位或位置寄存器窗口

“P[*]”位置类型有 P[i]与 PR[i]两种，可以把 P[i]理解成局部变量，即每个程序中的 P[i]只能在该程序中使用，其他程序使用不了，把 PR[i]理解成全局变量，所有程序都可共用 PR[i]。根据实际需要选择使用 P[i]时，参数 i 只能直接指定；使用 PR[i]时可以间接指定，参数 i 可以间接指定时，PR[i]中的 i 可以是 MR 寄存器，例如

```
MOVE PR[MR[1: ]], 500mm/s, FINE
```

就是间接指定的方式。速度参数也可以用 MR 寄存器指定，运动指令编辑界面点击速度参数对其进行修改，点击定位类型参数“FINE”可以修改定位类型。

新建位姿



保存为模板

点击“保存为模板“后在”常用指令“和”运动指令“界面都会有下图画面



图 4.17 运动指令模板窗口

点击快速编辑中的运动指令语句可在程序中快速插入运动指令

4.1.6 修改运动指令

在程序编辑界面选中需要修改的运动指令，被选中的语句行前会有光标显示，如图 4.12 所示。

点击“编辑指令“后出现下面画面。



图 4.17 运动指令编辑窗口

参考 4.1.5 节创建动作指令的步骤来修改运动指令。

4.1.7 创建动作附加指令

结合上节内容，在运动指令编辑界面点击运动指令后面的“+”后出现下图画面



图 4.18 附加指令编辑窗口

选择需要的附加参数，关于附加参数的说明参考 3.3.5 节

4.1.8 插入控制指令

控制指令，是除了动作指令外对在机器人上所使用的程序指令的总称。

插入 IF 指令

在程序编辑界面点击“插入指令”后，进入指令选择界面，然后点击“常用指令”或“逻辑指令”如下图。



图 4.18 插入指令窗口

选则 IF，进入 IF 语句编辑界面，如图 4.19 所示，点击“*”处，填加相应参数填写完成后，点击“确认”即可完成 IF 语句的添加。参数的添加见 IF 语句补充说明。



图 4.19 插入指令窗口

IF 语句编程示例如下图 4.20 所示：

```
2 IF R[1: default] = R[2: default]
3   R[2: default] = 1
4 ELIF R[2: default] = R[3: default]
5   R[2: default] = 2
6 ELSE
7   R[2: default] = 3
8 END IF
```

图 4.20 IF 指令程序示例

IF 语句补充说明

在 IF 语句编辑界面，可在 IF、ELSE IF、ELSE 之间进行切换，如下图所示



图 4.21 IF 指令切换界面

在插入 ELSE IF、ELSE 之前必须要插入 IF，否则 ELSE IF、ELSE 不会插入成功，还会有错误提示。



点击图 4.22 IF 语句界面中的“*”会显示下 IF 语句参数编辑界面，进行判断条件的添加。



图 4.22 IF 语句参数编辑界面

参数 1 可选的参数有寄存器和 IO，其中寄存器类型有数值寄存器 R[i]、IO 类型有专用 IO，数字 IO

参数 2 可选参数有寄存器、IO、IO 状态，其中其中寄存器类型有数值寄存器 R[i]、IO 类型有专用 IO，数字 IO，IO 状态有 ON、OFF。

元素类型

参数 1 或参数 2 可选的寄存器类型和 IO 类型

元素列表

参数 1 或参数 2 可选的寄存器类型和 IO 类型的具体元素，该列表中只显示已经创

建的或已经配置的元素

IF 语句详细界面中的符号说明

	代表增添一个表达式
	代表去除一个表达式（至少要有一个，即等式右边至少要有一项）
	代表增加括号
	代表去除括号

在 IF 语句编辑点击“=”可更换算符，可更换的算符有哪些及使用方法，请参考 3.8.11 复合运算指令

插入 SWITCH 指令

请参考 IF 指令的插入方法

插入 WHILE 指令

请参考 IF 指令的插入方法

插入寄存器及 IO 指令

关于寄存器指令的说明参见 3.4 节

插入数值寄存器-R[I]

在程序操作界面点击“插入指令”进入指令选择界面→点击“赋值及 IO”，切换到“赋值及 IO”指令界面，如图 4.23 所示。



图 4.23 赋值及 IO 指令界面

点击“R”，进入 R 寄存器指令参数填写界面，如图 4.24 所示

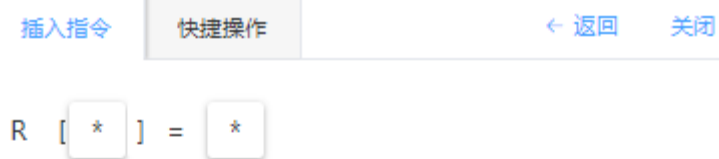
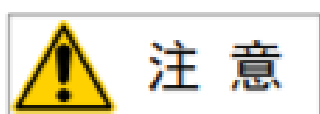


图 4.24 赋值及 IO 指令界面

点击“*”填写参数，参数填写完成，点击确认即可完成指令的插入。关于 R 寄存器的参数说明，详见 3.4.1 节



其他寄存器与 IO (DO/RO) 和 R 寄存器的插入步骤类似，请参考寄存器的插入步骤。

插入 WAIT 指令

在程序操作界面点击“插入指令”进入指令选择界面→点击“结构指令”，切换到“结构指令”界面，如图 4.25 所示。



图 4.25 结构指令界面

点击“WAIT”，进入 WAIT 指令参数填写界面，如图 4.26 所示。

插入指令

快捷操作

← 返回 关闭

WAIT DI[*]=ON +

确认

取消

图 4.26 WAIT 指令界面

点击“DI[*]=ON”处设置参数，参数设置完成，点击确认即可完成指令的插入。

补充说明

插入 WAIT 指令时，
默认的表达式是

WAIT [DI[*]=ON] +

，点击“DI[*]=ON”进入图 4.27 界面，在此界面可对表达式进行修改。

插入指令

快捷操作

← 返回

关闭

WAIT DI[*]=ON +

条件 DI [*] = ON



图 4.27 WAIT 指令界面

若想对上图中表达式中“=”左边的元素类型进行更改，点击“DI”处，可选择的元素类型有 IO 类型和寄存器类型。IO 类型包含的有 (DI/DO /UI/UO)，寄存器类型包含的有 (R/MI/MH)，如图 4.28 所示。



图 4.28 WAIT 指令参数界面

表达式中“=”右边元素类型的值根据“=”左边的元素类型而变化。设置完成后，点击“确认即可”IO类型的取值参考 3.5 节，寄存器类型的取值参考 3.4 节。

插入 WAIT TIME 指令

在程序操作界面点击“插入指令”进入指令选择界面→点击“结构指令”，切换到“结构指令”界面，如图 4.25 所示。

点击“WAIT TIME”，进入 WAIT TIME 指令参数填写界面，如图 4.29 所示。



图 4.29 WAIT TIME 指令参数填写界面

点击“0 sec”处设置参数，参数设置完成，点击确认即可完成指令的插入。

补充说明

在图 4.29 界面中点击“0 sec”后的界面如图 4.30 所示。

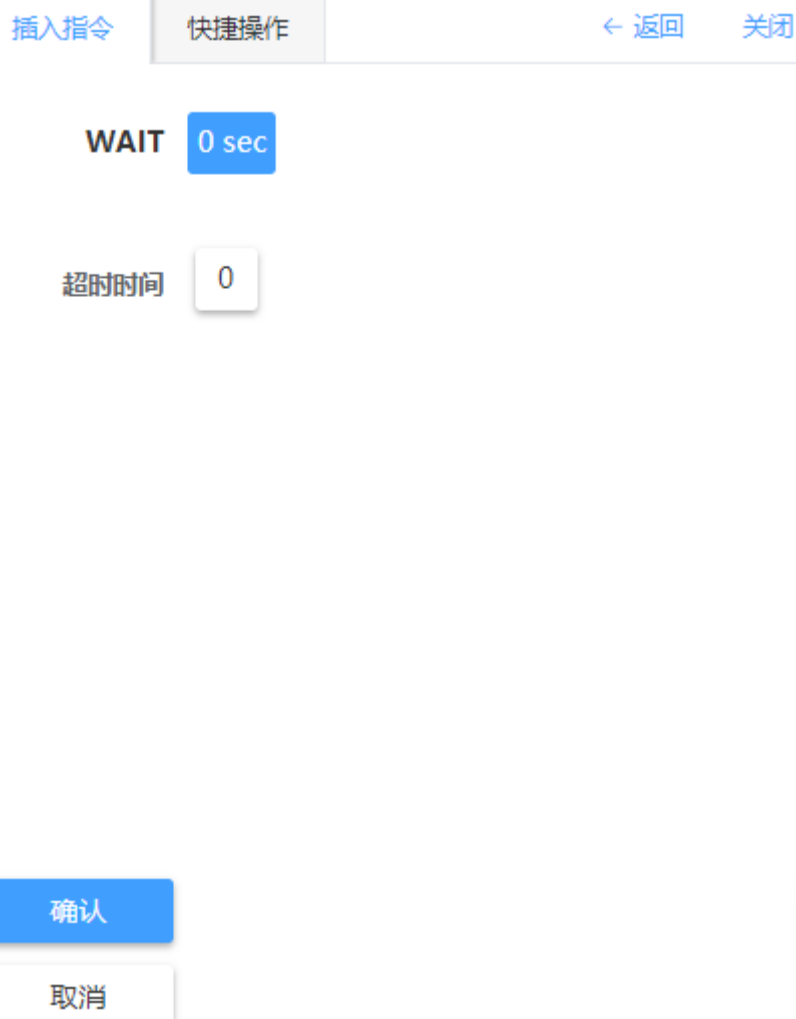


图 4.30 WAIT TIME 指令参数填写界面

在图 4.30 中点击“0”处设置超时时间，超时时间的数据类型有常数和数值寄存器 (R) 两种。

插入 LOAD 指令

在程序操作界面点击“插入指令”进入指令选择界面→点击“结构指令”，切换到“结构指令”界面，如图 4.25 所示。

点击“LOAD”，进入 LOAD 指令参数填写界面，如图 4.31 所示。

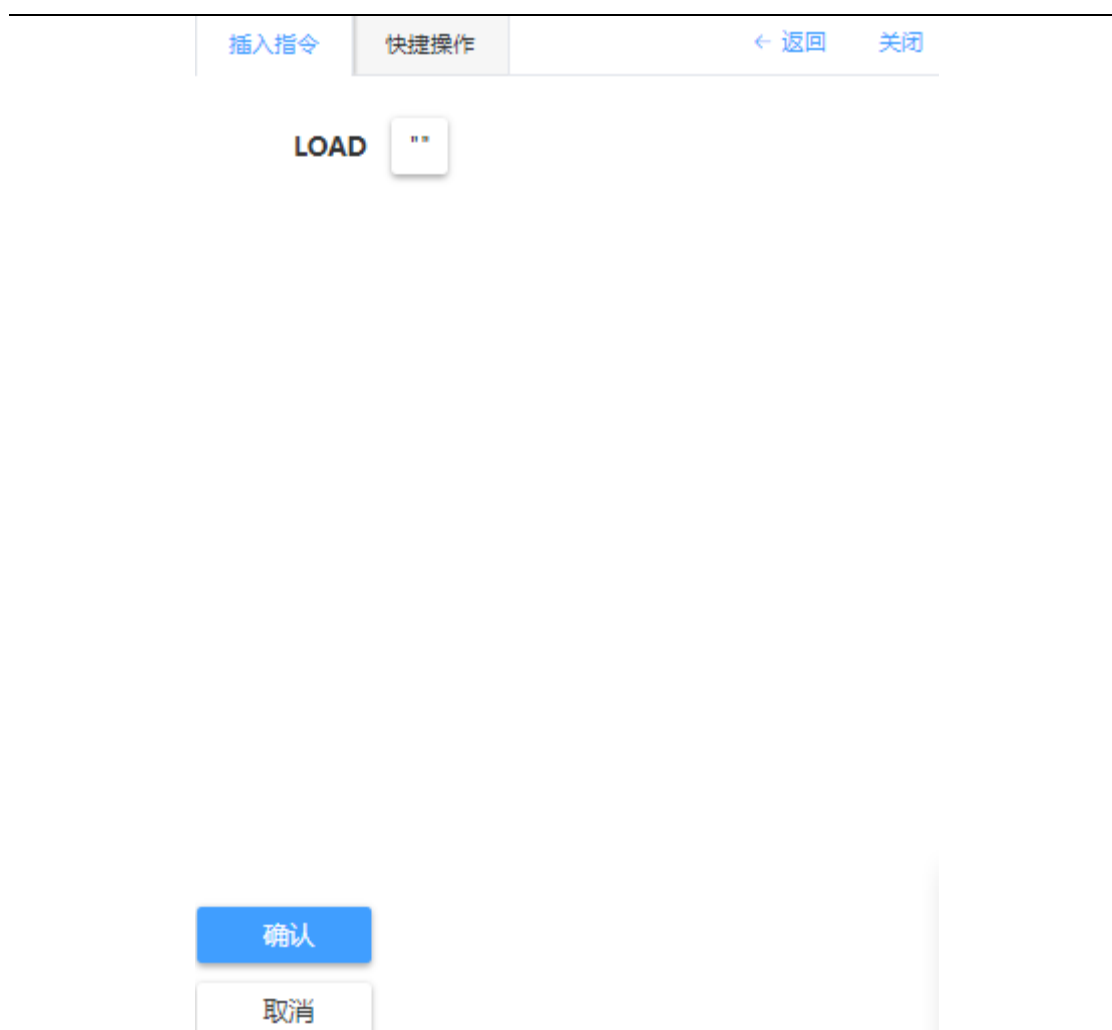


图 4.31 LOAD 指令参数填写界面

点击双引号处设置参数，参数设置完成，点击确认即可完成指令的插入。

补充说明

点击 LOAD 后面跟的“”可选择的数据类型有寄存器类型（R,SR 两种）、字符串、常数。

插入 EXEC 指令

参考 LOAD 指令的插入。

插入 UNLOAD 指令

参考 LOAD 指令的插入。

插入 PAUSE 指令

在程序操作界面点击“插入指令”进入指令选择界面→点击“结构指令”，切换到“结构指令”界面，如图 4.25 所示。

点击“PAUSE”，即可完成 PAUSE 指令的插入。

插入 ABORT 指令

参考 PAUSE 指令的插入。

插入 CALL 指令

在程序操作界面点击“插入指令”进入指令选择界面→点击“结构指令”，切换到“结构指令”界面，如图 4.25 所示。

点击“CALL”，进入 CALL 指令参数填写界面，如图 4.32 所示。

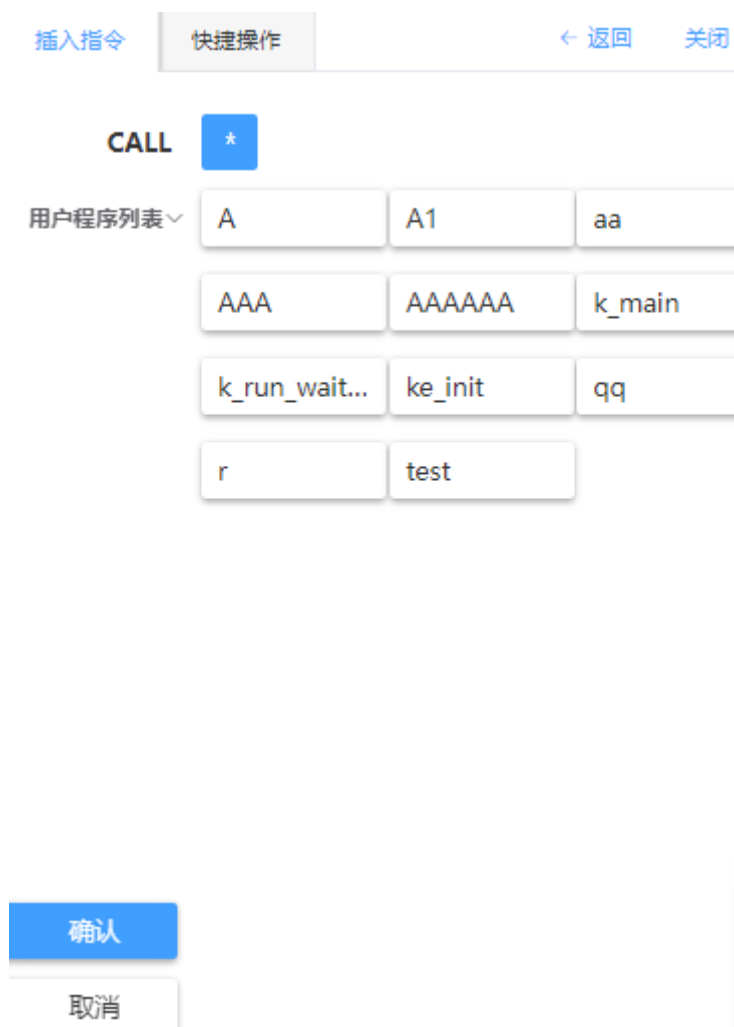


图 4.32 CALL 指令参数填写界面

在图 4.32 界面选择想要调用的程序后点击“确认”即可完成 CALL 指令的插入。

插入 SOCKET OPEN 指令

在程序操作界面点击“插入指令”进入指令选择界面→点击“SOCKET”，切换到“SOCKET”界面，如图 4.33 所示。



图 4.33 socket 指令界面

点击“SOCKET OPEN”，进入 SOCKET OPEN 指令参数填写界面，如图 4.34 所示。



图 4.34 SOCKET OPEN 指令参数填写界面

点击“SK[*]”选择套接字设备；点击“+”添加附加参数（返回参数），以上设置完成后，点击“确认”即可完成 SOCKET OPEN 指令的插入。

其他 SOCKET 指令的插入，请参考 SOCKET OPEN 指令的插入

插入 TF_NO.指令

在程序操作界面点击“插入指令”进入指令选择界面→点击“其他指令”，切换到“其他指令”界面，如图 4.35 所示。



图 4.35 其他指令插入界面

点击“TF_NO.”，进入 TF_NO.指令参数填写界面，如图 4.36 所示。



图 4.36 TF_NO.指令参数填写界面

在图 4.36 界面点击“*”设置工具坐标系号，工具坐标系号设置完成后，点击“确认”即可完成 TF_NO 指令的插入。

UF_NO.与 Payload_NO.和 OVC 指令的插入请参考 TF_NO.指令的插入。

插入 Comment 指令

在程序操作界面点击“插入指令”进入指令选择界面→点击“其他指令”，切换到“其他指令”界面，点击“Comment”，进入 Comment 指令参数填写界面，如图 4.37 所示。



图 4.37 Comment 指令参数填写界面

在图 4.37 中的长方形条框中填写注释数据，填写完成，点击“确认”即可完成 Comment 指令的插入。

插入 TIMER 指令

在程序操作界面点击“插入指令”进入指令选择界面→点击“其他指令”，切换到“其他指令”界面，点击“TIMER”，进入 TIMER 指令参数填写界面，如图 4.38 所示。

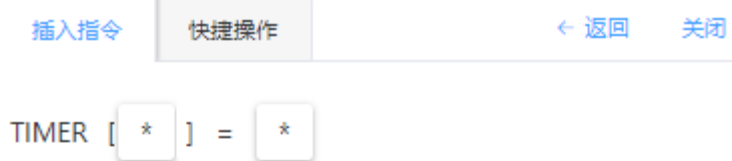


图 4.38 TIMER 指令参数填写界面

在图 4.38 中点击“*”处，设置计时器号码和计时器值（计时器相关参数请参考 3.8.6 节），设置完成，点击“确认”即可完成 TIMER 指令的插入。

插入 OAC-全局加速度指令

在程序操作界面点击“插入指令”进入指令选择界面→点击“其他指令”，切换到“其他指令”界面，点击“OAC”，进入 OAC 指令参数填写界面，如图 4.39 所示。

插入指令

快捷操作

← 返回

关闭

Overall_Acceleration_Coefficient = *

确认

取消

图 4.39 OAC 指令参数填写界面

在图 4.39 中点击“*”处，设置参数值，设置完成，点击“确认”即可完成 OAC 指令的插入。

插入 LABEL 指令

在程序操作界面点击“插入指令”进入指令选择界面→点击“其他指令”，切换到“其他指令”界面，点击“LABEL”，进入 LABEL 指令参数填写界面，如图 4.40 所示。

插入指令

快捷操作

← 返回 关闭

LABEL

1

ID

1

名称

确认

取消

图 4.40 LABEL 指令参数填写界面

在图 4.40 中 ID 后面的参数填写框处，设置 ID 号，设置完成，点击“确认”即可完成 LABEL 指令的插入；其中 ID 是标签号，名称处可以给标签命名。

插入 GOTO 指令

在程序操作界面点击“插入指令”进入指令选择界面→点击“逻辑指令”，切换到“逻辑指令”界面，点击“GOTO”，进入 GOTO 指令参数填写界面，如图 4.41 所示。

插入指令

快捷操作

← 返回

关闭

GOTO LABEL[2]

确认

取消

图 4.41 GOTO 指令参数填写界面

在图 4.41 红色框中，填写 LABEL 标签 ID 号，填写完成，点击“确认”即可完成 GOTO 指令的插入。

插入 SKIP CONDITION 指令

在程序操作界面点击“插入指令”进入指令选择界面→点击“逻辑指令”，切换到“逻辑指令”界面，点击“SKIP CONDITION”，进入 SKIP CONDITION 指令参数填写界面，如图 4.42 所示。



图 4.42 SKIP CONDITION 指令参数填写界面

在图 4.42 点击位置 1 处出现条件参数设置界面，位置 2 处可选的参数类型有 IO 类型和寄存器类型，位置 3 处参数类型有寄存器类型、IO 类型、数值、IO 状态；点击位置 2 与位置 3 之间的“=”可选择需要的操作符与布尔操作符；参数设置完成后，点击“确认”即可完成 SKIP CONDITION 指令的插入。

插入 ConllisionDetect 指令

在程序操作界面点击“插入指令”进入指令选择界面→点击“参数设置”，切换到“参数设置”界面，点击“ConllisionDetect”，进入 ConllisionDetect 指令参数填写界面，如图 4.43 所示。

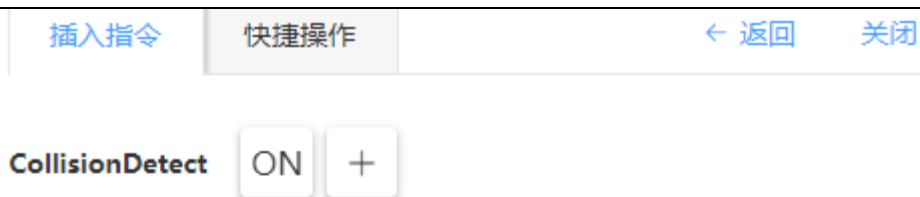


图 4.43 ConllisionDetect 指令参数填写界面

在图 4.43 中点击位置“+”可添加附加参数 Group 与 Axis，点击“ON”可切换碰撞检测状态为 ON 或 OFF，参数设置完成后，点击“确认”即可完成 ConllisionDetect 指令的插入。

插入 ConllisionRange 指令

在程序操作界面点击“插入指令”进入指令选择界面→点击“参数设置”，切换到“参数设置”界面，点击“ConllisionRange”，进入 ConllisionRange 指令参数填写界面，如图 4.44 所示。

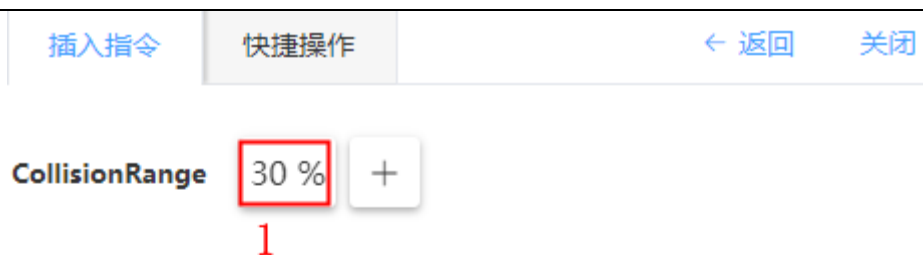


图 4.44 ConllisionRange 指令参数填写界面

在图 4.44 中点击位置“+”可添加附加参数 Group 与 Axis，点击位置 1 处可设置偏差范围值，参数设置完成后，点击“确认”即可完成 ConllisionRange 指令的插入。

4.2 程序设置

依次点击“菜单按钮”→“程序”进入界面如图 4.45 所示，“程序启动方式设定”设置。



4.2.1 程序启动方式设置

操作模式为自动模式下时，程序有四种启动方式：本地启动、主程序号启动、宏启动、主程序号启动简易模式。

用户可以通过依次点击“菜单按钮”-“程序”-“程序启动方式设定”进入程序启动模式设置界面，如图 4.47 所示，指定这 4 种启动方式中的某一种来启动机器人程序。系统默认方式是“本地启动”。



图 4.47 程序启动模式设置界面

本地启动

本地启动是指用户选择并打开希望启动的程序后（打开即进入程序编辑界面），点击 TP 上的开始按钮，该程序从第一行开始正序连续运行。

“本地启动”适用于所有类型程序。

“本地启动”的前提条件

- 机器人操作模式为“自动模式”
- 机器人运行状态处于“On-Standby”

- 程序执行状态为“Finished”
- “程序启动模式”设置为本地启动
- 有一个可执行的程序被加载（打开进入到程序编辑界面）

主程序号启动 (MPLCS)

主程序号选择启动，是通过寻找主程序的启动号进行启动。

主程序的启动号是十进制的。

主程序选择启动的作用流程如下：

1. 当启用“主程序号选择启动”后，系统收到上位机 Selection Strobe 信号 (UI[6]) 保持高电平。
2. 只要 UI[6]信号维持高电平，系统就会读取 UI[8]- UI[13]的 6 个 UI 信号的电平状态 (0/1) 来组成 6bits 的二进制数。
3. 系统还会通过 6 个 UO (Selection Confirm) 反馈自己读到的 6bits 的二进制数，并维持 Selection Check Request (UO[6]) 高电平信号给上位机进行握手请求。
4. 上位机收到 UO[6]检查 6 个 UO 的输出是否和给定值一致，如果一致，则发出 MPLCS Start 给机器人系统 (UI[7]) 并在一定时间之后断开（该时间必须大于 100ms）。
5. 机器人系统获取 UI[7]消失后，转化 6bits 的二进制数为十进制数，并依据该数字去寻找主程序号与之相等的主程序。（如果没找到则反馈报警事件）。
6. 找到要启动的主程序后，启动该程序。同时发出 UO[7]的 (MPLCS Start Done) 给上位机。
7. 上位机关闭 UI[6]、UI[8]- UI[13]信号。
8. 机器人系统一旦发现 UI[6]信号消失，复位 UO[6]、UO[8]- UO[13]、UO[6]信号。



目前考虑到标配 IO 板的 IO 数量限制，用于程序号启动的 UI Program Selection、UO Selection Confirm 都只有 6 位，即只能组成 6Bits 的启动代码。

具体时序可参考下图 4.48:

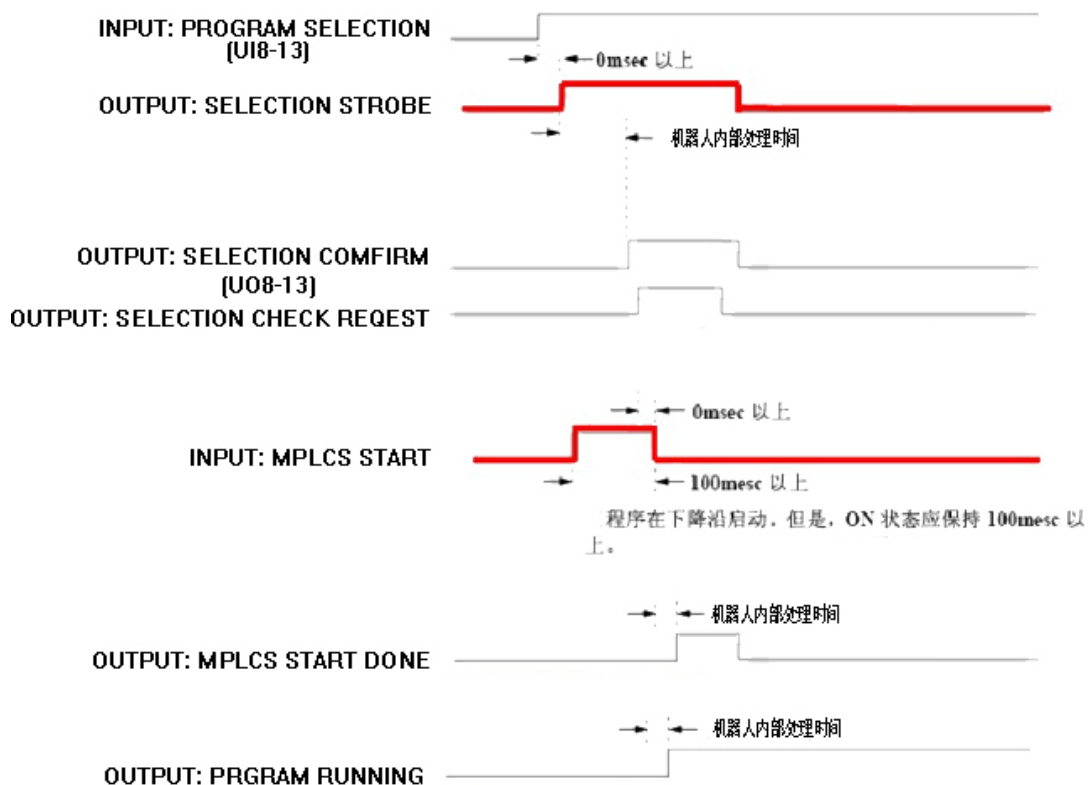


图 4.48 MPLCS 时序图

“主程序号启动”适用的程序类型是程序属性为 Main 的主程序。

“主程序号启动”的前提条件是

- 机器人操作模式为“自动模式”
- 机器人运行状态处于“On-Standby”
- 程序执行状态为“Finished”
- “程序启动模式”设置为主程序号启动

主程序号启动简易模式启动

主程序号启动完整模式流程比较复杂，很多场景下无需这样复杂的过程（普通集成用户也不希望时序逻辑太复杂）。所以提供一种程序号启动的简易模式。相对于完整模式，简易模式下简化了将收到的 UI[8]-UI[13]信号反馈给外部确认的相关过程。

- 当机器人处于“主程序号启动简易模式”，并满足对应启动的前提条件时，会检测 START 信号。
- 当 START 信号出现了下降沿(从高电平变为低电平)，即认为需要启动程序；然后去读取 UI[8]1~UI[13]，组合成一个 10 进制数作为程序码。尝试去找对应的程序启动。没找到则报警。

- 其他如暂停、停止等机制与完整模式无异。
主程序号启动简易模式适用的程序类型是程序属性为 **Main** 的主程序。
“主程序号启动简易模式”的前提条件是
- 机器人操作模式为“自动模式”
- 机器人运行状态处于“On-Standby”
- 程序执行状态为“Finished”
- “程序启动模式”设置为主程序号启动简易模式

补充说明

1. 当程序类型选择 **Main**，程序属性界面会增加“程序起始点”和“程序号”的功能，并且程序员权限可以选择是否禁用“程序起始点”功能。如图 4.49 所示。详情请参见 4.3 节。
2. 其中程序号：范围是 1-255，但由于用于调用 **Main** 程序的系统信号只有 6 个，6 个信号组成的最大十进制数是 64，所以外部调用 **Main** 程序的最大程序号是 64。

程序属性

程序适配机型: GBT-P7A-700×

程序名称	<input type="text" value="k_run_waitTrigger"/>
程序备注	<input type="text" value="程序备注支持不超过1000个字节的内容。"/>
程序类型	<input type="text" value="Main"/>
程序起始点	<input checked="" type="checkbox"/> 启用 <input type="checkbox"/> 未设置 <small>主程序需要设置程序起始点才能自动启动。</small>
程序号	<input type="text" value="1"/> <input type="text" value="0b00000001"/>

创建时间: admin 创建于 2023-02-07 14:08:42
修改时间: admin 修改于 2023-02-08 20:31:11

图 4.49 程序属性窗口

3. **None** 一般程序：没有什么特殊限制或特殊性。在自动模式时，无法通过除了本地启动方式(手动按启动按钮)以外的方式启动起来。但是可以被其他程序调用。
4. **Macro** 宏程序：在程序执行时，宏程序是能通过宏启动方式启动的程序。
5. 宏程序启动方式简述：用户在特殊程序设置（宏设置）界面设置，每个宏程序和一个 DI 一一对应。启动宏程序时可以通过预先设置好的 IO 进行启动。
6. 想通过“主程序号启动”或“主程序号启动简易模式”启动主程序，示教器模式开关必须在自动模式。

4.2.2 程序起始点

防止在自动模式下，程序自动启动时，由于机器人当前位姿不在程序员计划中的程序起始点，规划出的第一条路径与调试时不同，不符合程序员的预期路线前往第一个目标点，从而导致不可预知的危险或伤害，因此用户可对自动运行的程序设置程序起始点即 Home 点。依次点击“菜单按钮”→“程序”→“程序起始点”进入界面如图 4.50 所示：

程序起始点的要素包含：

- **J1-6** 是该点位的关节信息。显示单位：角度。
- **ID**：该点位的 ID 号，可以修改。
- **Comment**：注释；支持 128 个字节。
- **Name**：名称；支持 32 个字节。
- **浮动值**：Home 点校验时的正负浮动范围。显示层单位：角度。允许填写范围为 999.999，默认值 0.5。

用户可新增或者删除程序起始点，通过页面下方的“示教记录”按钮，可记录当前点为程序起始点。点击“编辑”按钮可更改机器人轴的值与浮动值。当用户想要快速示教机器人至已记录的程序起始点时，可以通过“移动到点”按钮，将机器人移动至该程序起始点位置。

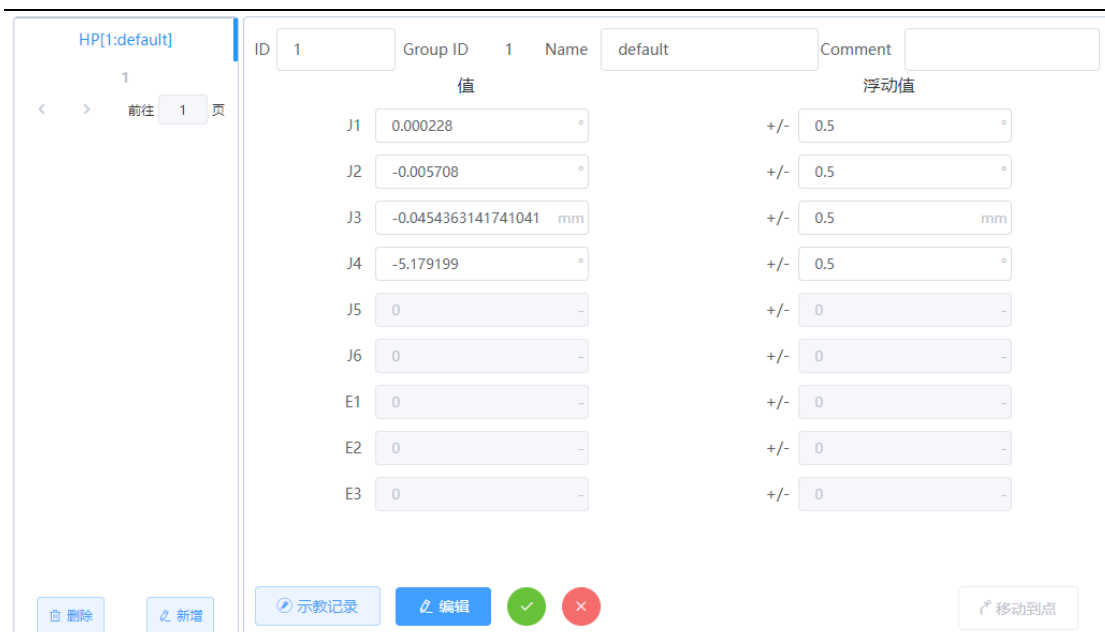


图 4.50 程序起始点窗口

4.3 执行程序

4.3.1 手动模式下启动程序

手动模式下执行程序也叫“程序调试执行”。机器人程序的执行需要按以下步骤操作：

1. 使示教器模式开关处于手动状态（手动全速或手动限速）
2. 打开需要执行的程序（打开程序参考 4.1.4 节）
3. 选择以何种方式执行程序（连续执行、单步正序、单步逆序）



图 4.51 程序执行方式选择

4. 选中需要从哪行开始执行程序的语句，选中后语句会被加粗，如下图所示

```

4 MOVEL P[3: ], 500mm/s, FINE
5 MOVEL P[5: ], 500mm/s, FINE
    
```

图 4.52 程序窗口

5. 点击“箭头指向光标”；启动程序时将会从选中行开始执行。

- 按下示教器反面使能按钮并保持在中间档位



图 4.53 使能按钮

- 按下示教器上的“reset”按钮，清除报警等信息并使机器人励磁（观察示教器状态指示灯“Servo ON 有没有亮绿灯”）
- 按下启动按钮，程序即运行

4.3.2 自动模式下启动程序

在自动模式下用户只可以连续地执行程序。

并且在自动模式下，程序必须通过系统设置规定的“程序自动启动方式”才能执行起来，或者从暂停中恢复。

只有当机器人运行状态栏的状态处于“ON-standby”（同时程序状态不处于“Paused”“Running”）时，才能通过自动模式的程序启动方式启动。

当自动模式下，如果想要恢复一个被暂停的程序使其继续被执行，只需要按下“启动”按键，或向控制柜发送“restart”的 UI 信号即可。

自动模式下程序的启动分为多种方式，详情请见 4.2.2 节。

机器人程序的执行需要按以下步骤操作：

- 打开需要执行的程序（通过程序号或宏指令调用不需要打开程序，只有手动执行时才需要打开相应程序）
- 方式开关选择自动模式（A）
- 按下示教器上的“reset”按钮或向控制柜发送“restart”的 UI 信号，清除报警等信息并使机器人励磁（观察示教器状态指示灯“Servo ON 有没有亮绿灯”）

4. 按照设定好的程序启动方式所对应的操作启动程序（直接按 TP 启动按钮、发规定好宏的 DI 触发信号、按主程序号启动方式规定的时序触发等），程序即运行

4.3.3 程序的暂停和终止

暂停

机器人在运行过程中，机器人的程序因某些原因暂停。暂停后，程序箭头停在正在执行的程序行处。

可以引起暂停的操作有：

- 程序运行时，用户点击暂停按钮；或机器人接收到一个外部 Pause 信号。
- 会发生引起“暂停”行为的报警，例如“STOP”、“Pause”、“Servo 1”等级报警。STEP 单步执行程序，或者逆序执行程序，每执行完一条指令后程序都会进入暂停状态。
- 程序执行过程中执行了一个“Pause”指令。



STOP、Servo 1 报警等级会引起运动的终止，但只会让程序暂停，比如触发安全门信号，会引起“STOP”报警，虽然运动会发生瞬间停止、但程序是变为暂停状态。

终止

两种终止状态：程序终止和运动终止

程序终止

程序终止：机器人在运行过程中，机器人程序可以被某些原因终止。终止后，系统维持着刚才正在执行的那一行指令序号。注意程序正常执行完或者执行了 Abort 指令，也认为是程序终止的一种特殊情况。

注意，被上级程序调用的子程序在执行过程中如果被终止，则返回上级程序的信息丢失，不能在再次独立启动并执行完毕后返回上级程序。

可以引起程序终止的操作有：

- 用户点击暂停按钮，再点击暂停按钮，界面会弹框 是否要停止当前程序，点击确认。
- 发生会引起“程序终止”行为的报警。
- 程序执行过程中执行了一个“END”指令。
- 程序执行过程中执行了一个“Abort”指令。

运动终止

运动终止：一般机器人运动终止也指的是机器人伺服电断开、抱闸抱紧。终止不区分此时机器人是否在运动、减速还是静止，直接关闭伺服电并抱闸。

一般的，当发生运动终止时，程序终止不一定也发生，可能会是程序暂停（比如按下急停按钮时）；程序终止也不一定引发运动终止，可能程序 END 了，机器人维持励磁，等待去命令执行另一个程序。

可以引起运动终止的操作有：

- 用户点击暂停按钮，再点击暂停按键，界面会弹框 是否要停止当前程序，点击确认。
- 发生会引起“运动终止”行为的报警。
- 程序执行过程中执行了一个“Abort”指令。

4.3.4 暂停后的恢复

当机器人的程序执行状态处于“暂停”时，如果机器人运行状态处于“On-Standby”，并且当前没有“Pause”级以上的报警在生效，用户就可以点击启动按钮进行程序的恢复启动。

4.4 程序监控

依次点击“菜单按钮”→“程序”→“程序监控”即可进入程序监控界面。

程序监控界面如下图所示

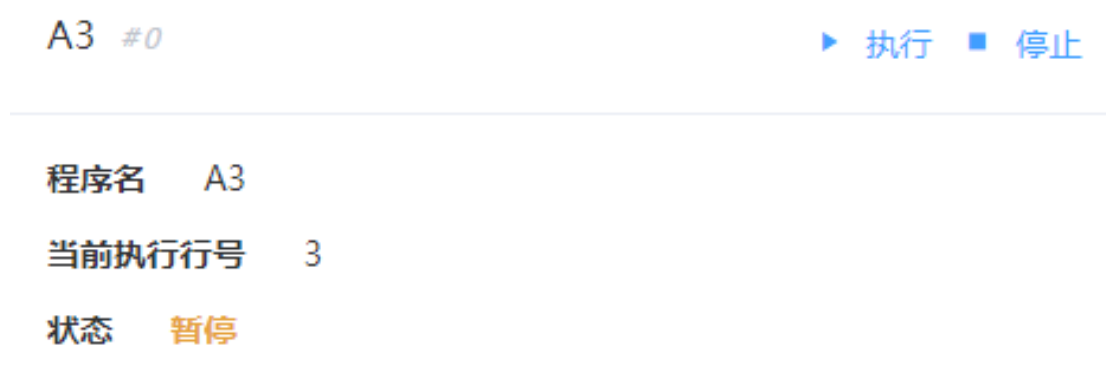


图 4.54 程序监控界面

点击“执行”可在“执行”与“暂停”之间切换，执行即是执行程序，暂停即是暂停程序，点击“停止”程序停止。

状态：

- 运行中
- 暂停
- 终止（随后程序会消失）



对于 python 脚本程序，暂停无效，终止有效。

5. 状态显示

5.1 寄存器管理

捷勃特机器人所用的寄存器有数值寄存器（R）、运动寄存器（MR）、位姿寄存器（PR）、字符串寄存器（SR）、Socket 寄存器、Modbus 寄存器。

5.1.1 数值寄存器

数值寄存器（R[i]）的显示和设置在数值寄存器画面上进行。

依次点击“菜单按钮”→“数据”→“数值寄存器”可进入数值寄存器设置界面，如图 5.1 所示。

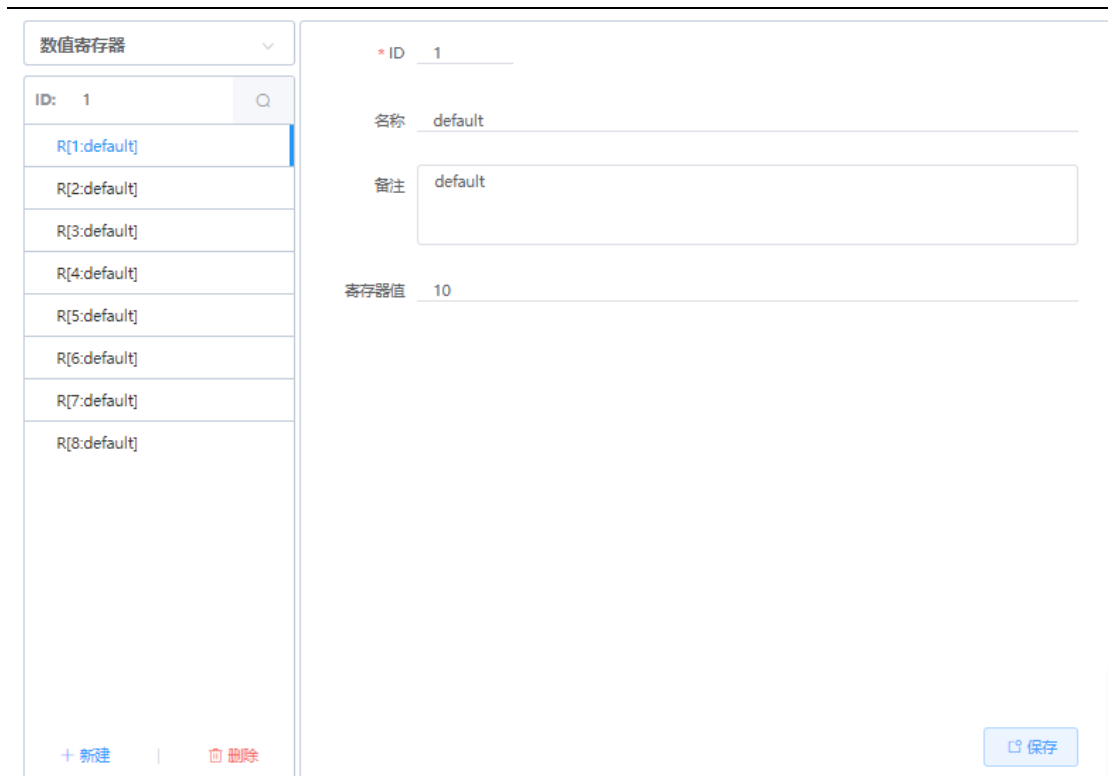


图 5.1 数值寄存器设置界面

在数值寄存器设置界面可以进行新建、删除寄存器操作。

数值寄存器参数说明

- ID:寄存器号码。
- 名称：在此处为数值寄存器命名。名称可以是不超过 31 字节的字符组成，但不允许出现：# \$ % ^ & * () @ + - = \ , ; : " ' | < > ~ { } 这些符号。
- 备注：在此处为数值寄存器添加备注，备注可以是不超过 128 字节的字符。
- 寄存器值：在此处设置寄存器值。范围是：范围为±99999999.999。

以上参数设置完成，点击“保存”即可。

5.1.2 运动寄存器

运动寄存器（MR[i]）的显示和设置在数值寄存器画面上进行。

依次点击“菜单按钮”→“数据”→“运动寄存器”可进入运动寄存器设置界面，如图 5.2 所示。

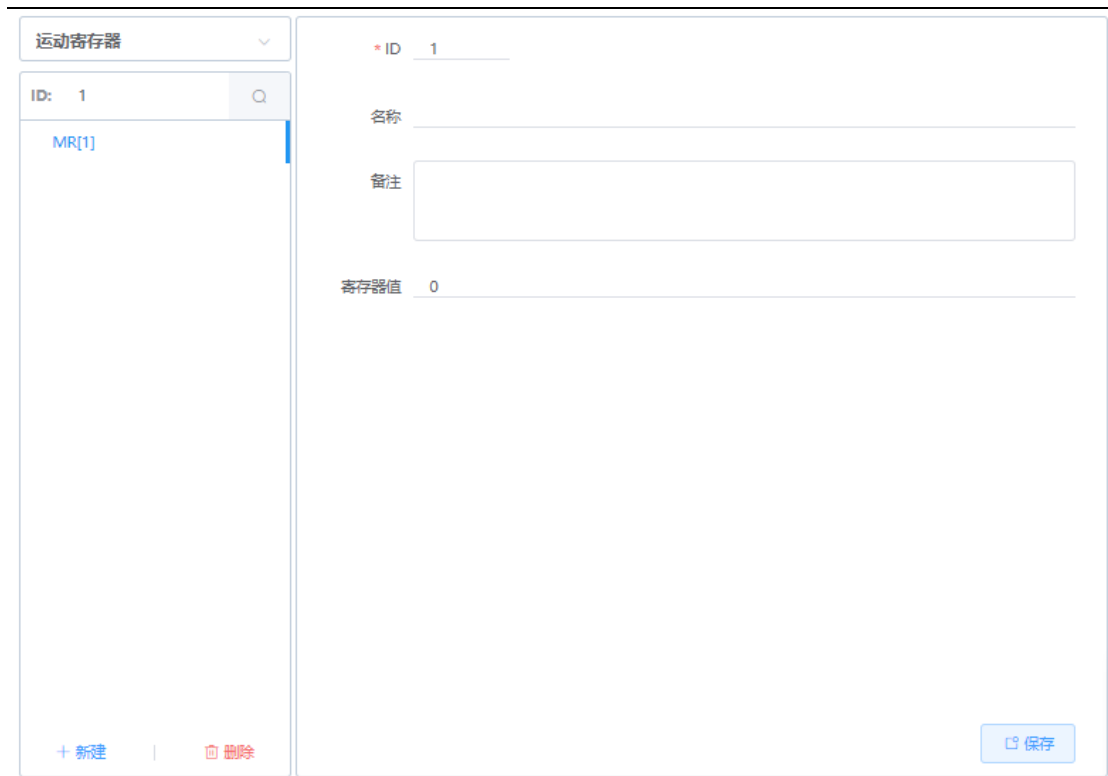


图 5.2 运动寄存器设置界面

在运动寄存器设置界面可以进行新建、删除寄存器操作。

运动寄存器参数说明

- ID:寄存器号码。
- 名称：在此处为运动寄存器命名。名称可以是不超过 31 字节的字符组成，但不允许出现：# \$ % ^ & * () @ + - = \ , ; : " ' | < > ~ { } 这些符号。
- 备注：在此处为运动寄存器添加备注，备注可以是不超过 128 字节的字符。
- 寄存器值：在此处设置寄存器值。范围是：范围为±99999999 之间的整数。

以上参数设置完成，点击“保存”即可。

5.1.3 位姿寄存器

位姿寄存器（PR[i]）的显示和设置在位姿寄存器画面上进行。

依次点击“菜单按钮”→“数据”→“位姿寄存器”可进入位姿寄存器设置界面，如图 5.3 和图 5.4 所示。

位姿寄存器 ▼

* ID 1 组ID 1 名称 default

ID: 1 Q

PR[1:default]

PR[2]

PR[3]

PR[4]

PR[5]

PR[6]

备注

Cart
Joint

* X 1 mm

* Y 0 mm

* Z 0 mm

* A 0 °

* B 3 °

* C 0 °

回转数
 J3 0 J4 0 J6 0

关节配置

NOFLIP
UP
FRONT

+ 新建
删除

示教记录
编辑

移动到点

图 5.3 位姿寄存器设置界面（笛卡尔）

位姿寄存器 ▼

* ID 1 组ID 1 名称 default

ID: 1 Q

PR[1:default]

PR[2]

PR[3]

PR[4]

PR[5]

PR[6]

备注

Cart
Joint

* J1 180.000 °

* J2 -133.710 °

* J3 -12.277 °

* J4 0 °

* J5 -121.013 °

* J6 0 °

+ 新建
删除

示教记录
保存
取消

移动到点

图 5.4 位姿寄存器设置界面（关节）

在位姿寄存器设置界面可以进行新建、删除寄存器操作；通过 Cart/Joint 进行笛卡

尔坐标系数据（图 5.3）与关节坐标系数据（图 5.4）的切换。

位姿寄存器参数说明

- ID:寄存器号码
- 组 ID: 数值“1”表示机器人本体，其他数值表示外部轴，现在仅支持数值“1”本体表示。
- 名称: 在此处为运动寄存器命名。名称可以是不超过 31 字节的字符组成，但不允许出现: # \$ % ^ & * () @ + - = \ , ; : " ' | < > ~ { } 这些符号。
- 备注: 在此处为运动寄存器添加备注，备注可以是不超过 128 字节的字符。
- 寄存器值: 在此处设置寄存器值。范围是: 范围为±99999999 之间的整数。
- 笛卡尔坐标系下的值 (X,Y,Z,A,B,C) 表示指定 TCP 在指定用户/基/世界坐标系下的位姿信息。
- 关节坐标系下的值 (J1,J2,J3,J4,J5,J6) 表示机器人各关节的角度值。

坐标值的写入

- 坐标值可以通过“示教记录”按钮可将当前机器人坐标值记录下来（该值不含用户坐标系和工具坐标系信息。所以，在程序中使用 PR 位姿寄存器时，需要通过 UF_No.和 TF_No. 指令提前指定需要使用的用户坐标系或者工具坐标系，不同的用户坐标系下或者工具坐标系下 PR 寄存器代表的空间位置不一定是相同的。），若没在程序中指定坐标系，则使用当前选择的坐标系。
- 还可以通过点击“编辑”按钮填入坐标值，坐标值填写完成后点击“保存”即可完成坐标值的填写。

手动模式并且伺服上电前提下，在位姿寄存器界面点击“移动到点”，机器人会移动到位姿寄存器中的点位。

5.1.4 字符串寄存器

字符串寄存器 (SR[i]) 的显示和设置在字符串寄存器画面上进行。

依次点击“菜单按钮”→“数据”→“字符串寄存器”可进入字符串寄存器设置界面，如图 5.5 所示。

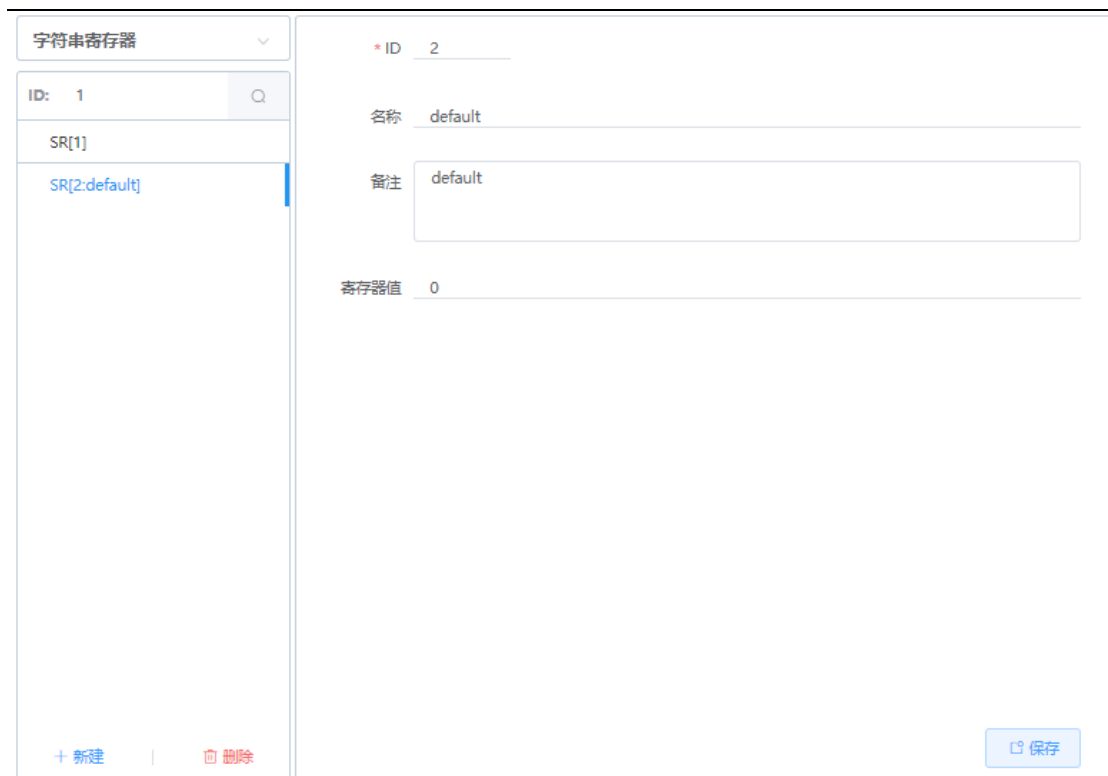


图 5.5 字符串寄存器设置界面

在字符串寄存器设置界面可以进行新建、删除寄存器操作。

字符串寄存器参数说明

- ID:寄存器号码。
- 名称：在此处为字符串寄存器命名。名称可以是不超过 31 字节的字符组成，但不允许出现：# \$ % ^ & * () @ + - = \ , ; ' " | < > ~ { } 这些符号。
- 备注：在此处为字符串寄存器添加备注，备注可以是不超过 128 字节的字符。
- 寄存器值：在此处设置寄存器值。最多不能超过 255 个字节。

以上参数设置完成，点击“保存”即可。

5.1.5 Socket 寄存器

与 socket 指令配合使用（socket 指令见 3.8.10 节）。

概述

机器人作为 Socket Client 或 Server 与其他设备进行 Socket 通讯。

操作流程：

在 Socket 寄存器页面，配置机器人作为 Client 或 Server 的参数，如通讯协议、

IP、port 等

编写程序时，调用 Socket 程序指令。

位置

依次点击“菜单按钮”→“数据”→“Socket 寄存器”进入 Socket 寄存器设置页面，如图 5.6 与 5.7 所示。功能协议，可选 TCP、UDP。机器人可做 Server 或 Client，对应配置项不同。

The screenshot displays the 'Socket 寄存器' (Socket Register) configuration interface. On the left, there is a search bar with 'ID: 1' and a list containing 'SK[1]'. Below the list are buttons for '+ 新建' (New) and '删除' (Delete). The main configuration area on the right includes the following fields:

- * ID: 1 (with a search icon)
- 名称: _____
- 协议: TCP (dropdown menu)
- 机器人做: Client (dropdown menu)
- 服务端 IP: 0 . 0 . 0 . 0
- * 服务端 Port: 1025
- 客户端 IP: 192 . 168 . 110 . 3
- * 接受超时默认时间: 15 s

A '保存' (Save) button is located at the bottom right of the configuration area.

图 5.6 Socket 寄存器设置页面（客户端）

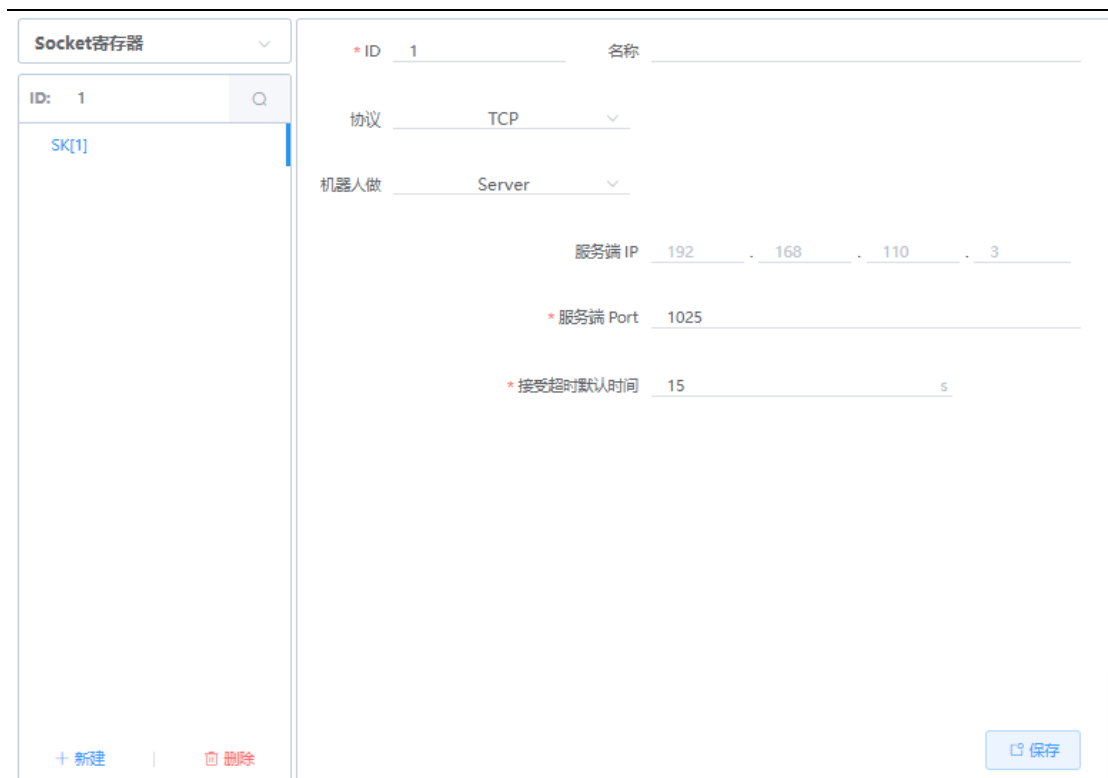


图 5.7 Socket 寄存器设置页面（服务端）

Socket 寄存器参数说明

- ID:寄存器号码。
- 名称：在此处为字符串寄存器命名。名称可以是不超过 31 字节的字符组成。

机器人做 Server（如图 5.7 所示）：

- 服务端 IP，表示机器人的 IP。
- 服务端 Port，表示该 Socket 使用的端口号，与 Client 连接时将使用该端口。
- 接受超时默认时间，表示当从 Socket 接受数据时的等待时间，若超过该时间，则不再等待，接受超时对应程序指令的返回值为 999。

机器人做 Client（如图 5.6 所示）：

- 服务端 IP，表示机器人所要连接的对端 Server 的 IP，由用户输入。
- 服务端 Port，表示机器人所要连接的对端 Server 的 Port，由用户输入。
- 客户端 IP，表示机器人的 IP（不可在图 5.7 所示界面更改）。
- 接受超时默认时间，同 Server 的定义。

5.1.6 Modbus 专用寄存器

用于监控 Modbus 寄存器，目前支持从站输入寄存器 Input Registers 及保持寄存器 Holding Registers 的监控。

其中，Modbus Holding Registers 以 MH[i]表示，Modbus Input Registers 以 MI[i]表示，其中 i 代表序号。

依次点击“菜单按钮”→“数据”→“Modbus 专用寄存器”进入 Modbus 专用寄存器设置页面，如图 5.8 所示。



进入 Modbus 专用寄存器页面前，Modbus 从站功能必须是激活的，否则进入 Modbus 专用寄存器页面会报错。

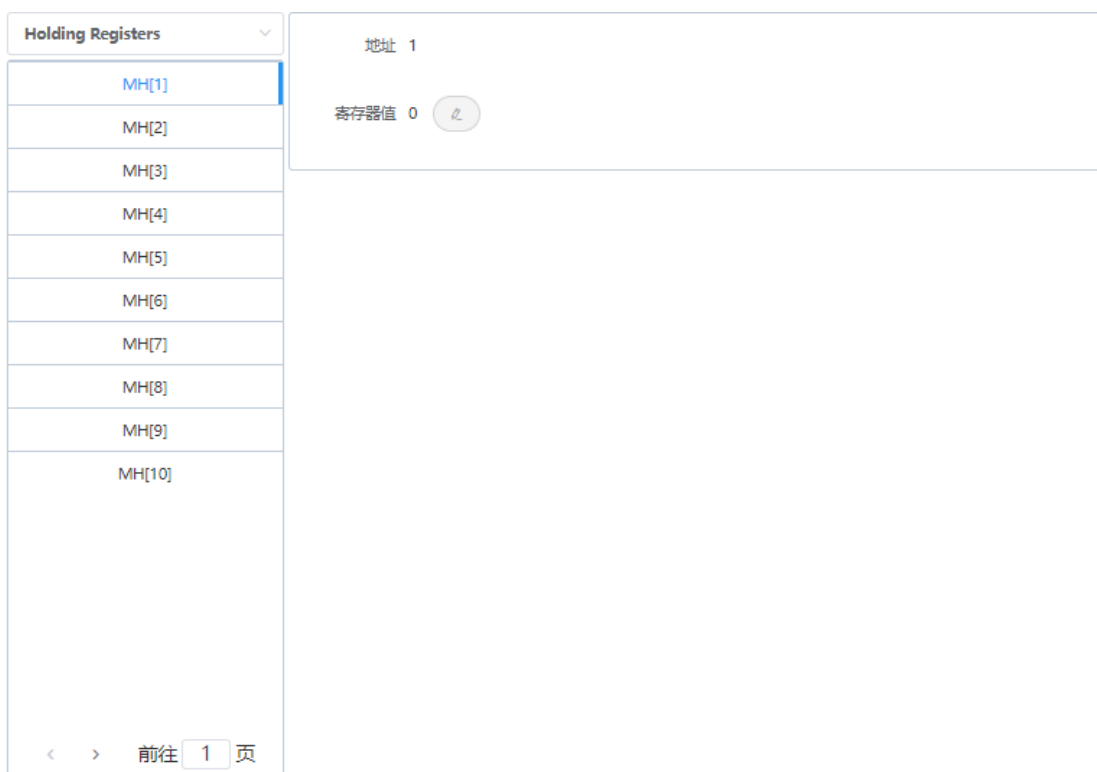


图 5.8 Modbus 专用寄存器设置页面

可以切换 Input Registers 及 Holding Registers 监控界面，如图 5.8 所示。

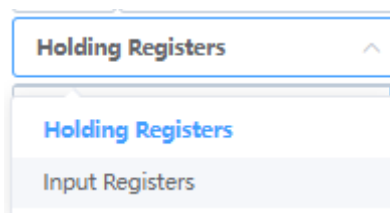


图 5.8 Modbus 寄存器类型切换窗口

IO 映射与 Modbus 专用寄存器

可利用 IO 映射将 Modbus 专用寄存器中的 Inputs/Coils 映射为 DI/DO，步骤如下：

1. 依次点击“菜单按钮”→“通讯”→“总线配置”进入图 5.9 界面。

The screenshot shows a configuration window for a Modbus slave. It includes the following fields and values:

- 从站ID (Slave ID): 1
- 名称 (Name): [Empty]
- 通信 (Protocol): Controller/TCP
- IP地址 (IP Address): 172.16.1.203
- 端口号 (Port): 502
- 扫描间隔 (Scan Interval): 0
- 数据类型 (Data Type): 无符号 (Unsigned)

A red '应用' (Apply) button is located at the bottom center of the form.

图 5.9 总线配置窗口

2. 进入 IO 映射页面，映射需要的 IO（Coil Status 映射给 DI，Discrete inputs 映射给 DO）（具体操作见 2.1.1 节）
3. 进入 IO 状态页面，监视及操作 DI/DO。

*关于 Modbus 更详细的配置和使用方式、实践案例，请查阅《捷勃特 Modbus TCP 功能手册》。

5.1.7 码垛寄存器

码垛寄存器是用于码垛功能的专用寄存器，系统配置中允许设置的码垛寄存器的最大个数为 30 个。码垛寄存器在码垛工艺工程生效（失效）时自动新建（删除），不可单独存在。码垛寄存器 1-30 分别对应 1-30 的码垛工艺工程。

码垛寄存器记为: PL[i:NAME]

数据模式包含 3 个元素: [R,C,L]

参数	描述
i	码垛寄存器的 ID 号，代表 1 至 30 的码垛工艺工程
NAME	码垛工艺名称，默认为 default

R	码垛行的数据, 其数据不可超出该码垛工艺工程的行配置。
C	码垛列的数据, 其数据不可超出该码垛工艺工程的列配置。
L	码垛层的数据, 其数据不可超出该码垛工艺工程的层配置。

即 PL[i] 中的 i, 范围 1 至 30, 即 PL[1] ~ PL[30] 代表 PALLETIZING[1]~PALLETIZING[30] 的码垛工艺工程。PL 寄存器是全局寄存器, 所有程序均可见。原则上是不允许 PL 寄存器在没有使用对应的码垛工艺工程的程序中单独操作; PL 寄存器的值断电后保存。

依次点击“菜单按钮”→“数据”→“码垛寄存器”可进入码垛寄存器设置界面, 如图 5.11 所示。

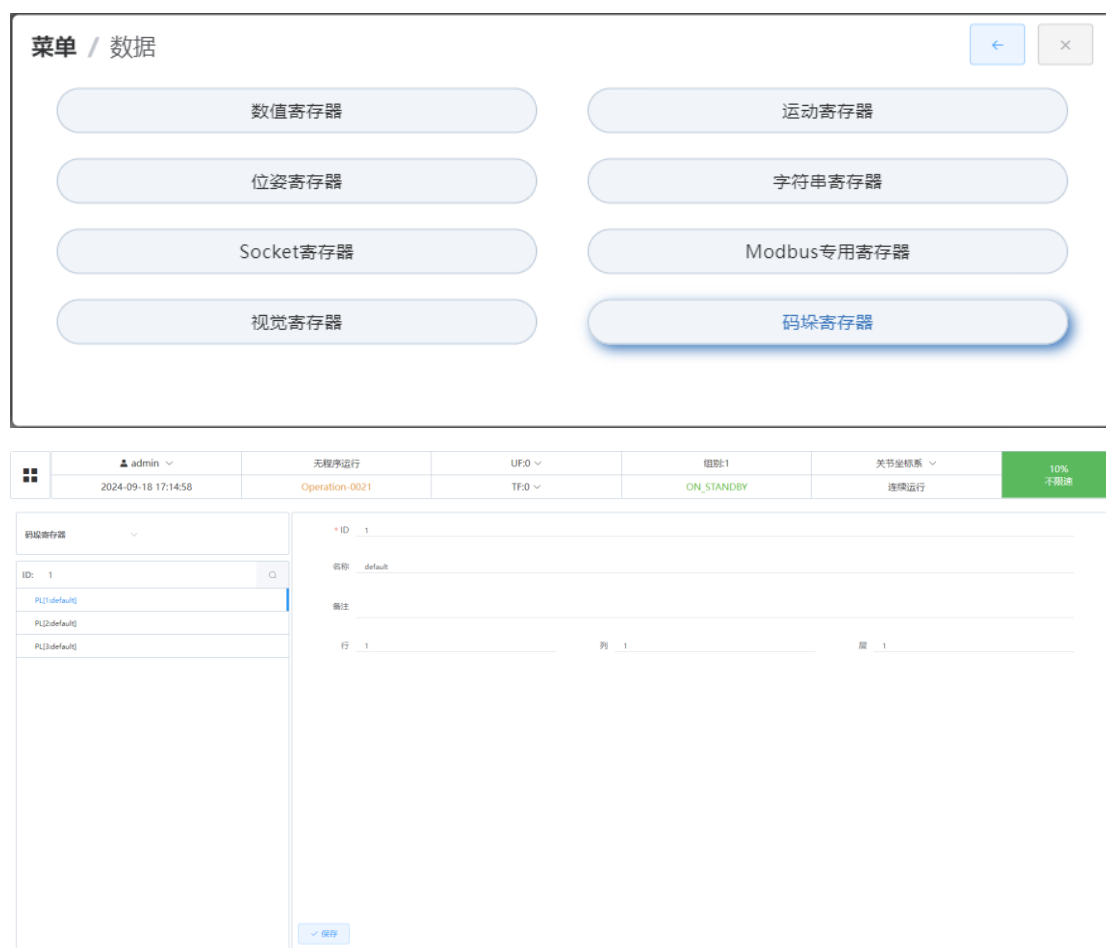


图 5.11 码垛寄存器设置界面

5.1.8 视觉寄存器

点击“菜单”→数据→视觉寄存器，进入图 3.1 界面。



图 3.1 VR 寄存器设置界面

基本信息

- ID 号：该寄存器的 ID
- 组 ID：该寄存器的运动组，默认为 1
- 名称：可备注该寄存器的名字，默认为空

视觉信息

- 模板 ID：检出工件的模板 ID
- 坐标系：补偿数据的坐标系编号
- 类型：检出工件坐标值的补偿类型
- 基准点偏移：基于用户坐标系补偿的数据进行定位纠偏，坐标系编号为用户坐标系
- 用户坐标系：检出工件的绝对位置，坐标系编号为用户坐标系

5.2 当前位置

为当前位置界面（如图 5.11）该界面主要是用于在机器人示教时实时的为用户提供机器人的当前位姿信息；以及为用户提供一个便捷移动到某确认目标点功能。

该界面由两部份组成，机器人当前位姿信息（图 5.11 区域 1）与目标点位（图 5.11 区域 2）。



图 5.11 当前位置界面

当前位姿信息区域说明

当前位姿信息显示机器人当前所处位姿信息，方便用户在示教过程中对机器人点位进行实时的观察。当系统坐标系选择为笛卡尔空间坐标系时，显示数据如图 5.11 所示；当选择关节坐标系时，显示数据如图 5.12 所示。

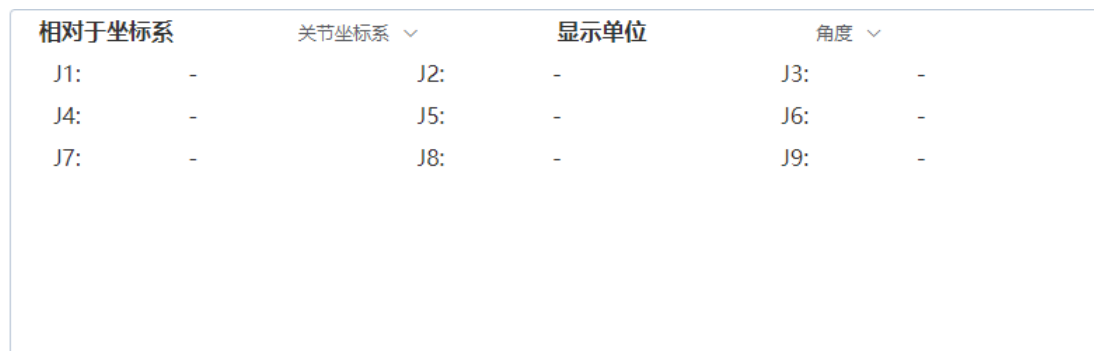


图 5.12 当前位置界面（关节）

相对于坐标系

关节坐标系

点击“相对于坐标系”选择关节坐标世界坐标系、基座坐标系、关节坐标系、用户坐标系。

- 若选择世界坐标系、基坐标系、用户坐标系时，则显示当前机器人的 TCP 在世界坐标系、基坐标系、当前用户坐标系下的坐标值。

- 若选择关节坐标系时，则显示当前机器人的关节坐标值。（与状态栏上的“当前坐标系选择”无关）。

显示单位角度 

点击“显示单位”，进行角度单位转换：角度制、弧度制。

“目标点位”区域说明



用户可以输入指定的目标点（可以输入笛卡尔坐标数据类型和关节坐标数据类型两种，点击 Cart/Joint 切换输入数据类型），目标点位数据填写完成后，机器人上使能上电，按住“移动到点”机器人将以关节运动方式（MoveJ）缓慢移动到目标点位。

点击“应用当前位姿”，可将机器人当前的位姿数据录入至目标点中。



移动过程中，用户需要一直按住“移动到点”按钮，直到机器人到达目标点。若中途松开了该按钮，机器人立即减速停止。

6. 机器人通信设置

概要

捷勃特机器人提供了 Modbus TCP 协议用于客户使用总线的方式与外围设备进行通信

Modbus TCP 是一种开放式主/从应用通讯协议，捷勃特机器人目前仅支持从站功能。其 4 种传输数据格式如下：

功能	类型	权限	Robot mapping
Coils 离散量输出（线圈）	Single bit 单个比特	读写	Digital Input
Discrete Inputs 离散量输入（线圈）	Single bit 单个比特	只读	Digital Output
Input Registers 输入寄存器	16-bits word 16 比特字	只读	MI 寄存器
Holding Registers 保持寄存器	16-bits word 16 比特字	读写	MH 寄存器

6.1 机器人作从站配置

依次点击“菜单按钮”→“通讯”→“总线配置”进入 Modbus 从站配置界面如图 6.1 所示。

The screenshot shows the Modbus Slave Configuration interface. At the top, there are tabs for 'Slave' (selected) and 'Master'. On the left, there is a status indicator '已停用' (Disabled) with a crossed-out robot icon. The main configuration area includes:

- 从站ID (Slave ID): 1
- 名称 (Name): [Empty field]
- 通道 (Channel): Controller/TCP
- IP地址 (IP Address): 192 . 168 . 110 . 2
- 端口号 (Port): 502
- 扫描间隔 (Scan Interval): 0
- 数值类型 (Numeric Type): 有符号 (Signed) 无符号 (Unsigned)
- 启用 (Enable) button: [Blue button with '启用' text]

图 6.1 当前位置界面

参数说明：

- 从站 ID: 从站（机器人）ID
- 名称: 从站名称

- 通道：Controller/TCP
- IP 地址：从站（机器人）IP 地址
- 端口号：与主站通讯所用端口
- 从站响应延迟：主站与从站通讯时规定的最迟响应时间
- 数据类型：有符号/无符号

从站设置步骤

1. 在图 6.1 中，配置 IP 地址必须与机器人在同一个域内（机器人默认 IP 为 192.168.110.2），名称和从站的响应延迟时间设置不可为空。
2. 菜单通讯→通讯映射进入寄存器设置页面，如图 6.2 所示

DI	DO	UI	UO	GI	GO	AI	AO	MH	MI	观察状态	新增	删除	
用户端口				模块号						地址	总数	状态	
1	MH	1	~	10	ModbusSlave/Controller/TCP/128						0	10	ACTIVE

图 6.2 总线寄存器信号设置页面

3. 如图 6.2 中根据实际需要设置寄存器地址及个数。
 - Coil 寄存器：地址和范围是 0-65536
 - Discrete Input 寄存器：地址和范围是 0-65536
 - Holding Register 寄存器：地址和范围是 0-9999
 - Input Register 寄存器：地址和范围是 0-9999
1. 寄存器地址及个数设置完成后，点击“配置”回到 6.1 界面，点击激活配置，如图 6.3 所示。



图 6.3 当前位置界面

*关于 Modbus 更详细的配置和使用方式、实践案例，请查阅《捷勃特 Modbus TCP 功能手册》。

6.2 机器人作主站配置

依次点击“菜单按钮→通讯→总线配置→主站”进入 Modbus 主站配置界面如图 所示。

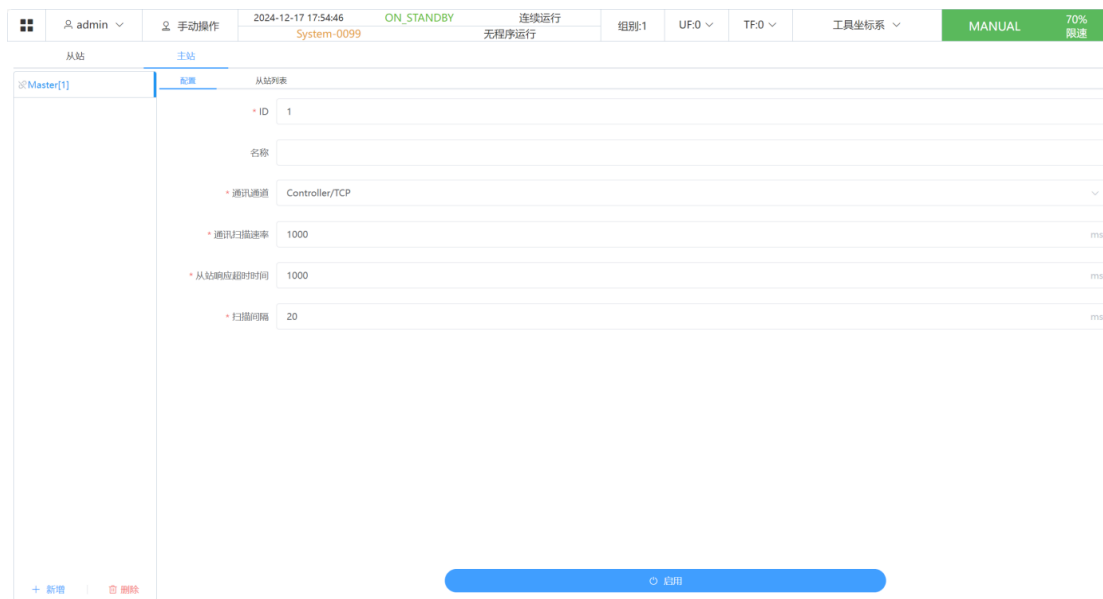


图 6.3 主站界面

参数说明：

- ID: ModbusID
- 名称: 最多 128 字节

- 通讯通道：可选 Controller/TCP、Controller/TCP2RS485 模式
- 通讯扫描速率：主站轮询周期，默认 1000 ms
- 从站响应超时时间：主站与从站通讯时，规定的最长响应时间
- 扫描间隔：主站轮询各从站的周期，默认 20ms



主站启用前，应保证个从站通讯正常，否则将启用失败。。

如使用 TCP，则应选择 Controller/TCP 模式，如图 6.4。



图 6.4 Controller/TCP 模式

如使用 TCP 转 RTU 网关，则应选择 Controller/TCP2RS485 模式，如图 6.5。此项需额外填写中继转换模块的 IP 地址和端口，模块应保证使用透传模式。



图 6.5 Controller/TCP2RS485 模式

在主站的从站列表，可创建该主站连接的 Modbus 从站，最多可建立 32 个从站。如下为选择 Controller/TCP 模式下的从站列表：

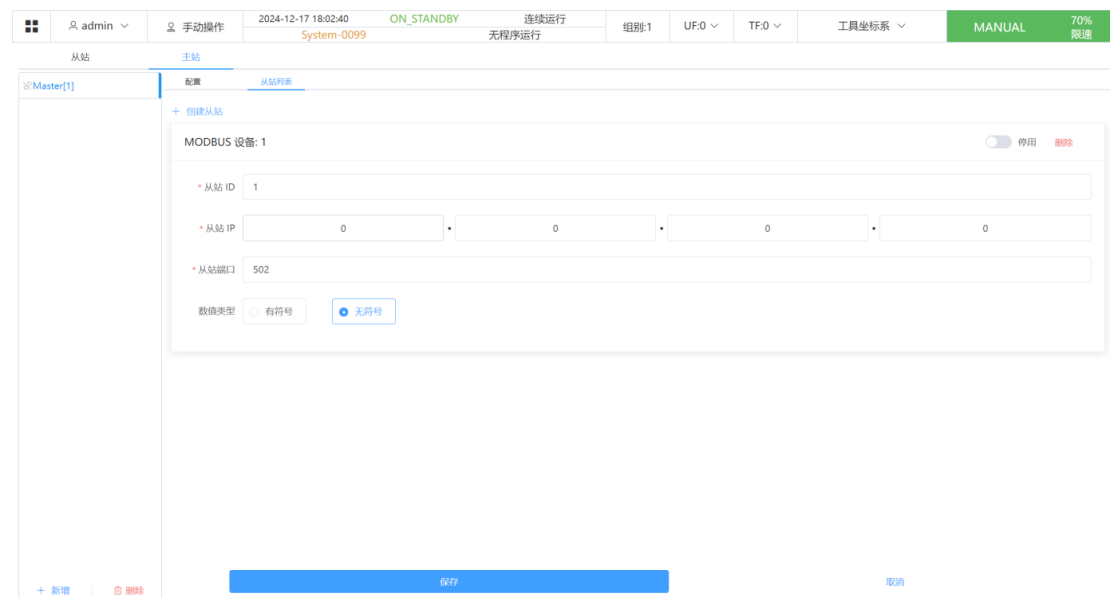


图 6.8 主站的从站列表

参数说明：

- 从站 ID: Modbus 从站 ID
- 从站 IP: 从站 IP 地址

- 从站端口：与主站通讯所用端口
- 数值类型：有符号/无符号

*关于 Modbus 更详细的配置和使用方式、实践案例，请查阅《捷勃特 Modbus TCP 功能手册》。

7. 系统备份与加载

- 防止程序或者软件在使用、传输中意外破损或丢失。
- 防止程序修改后认为原始程序较好要加载恢复。
- 机器人软件、程序及文件的整体覆盖或加载。

7.1 可以使用的储存设备

以 USB-A 为接口，USB2.0 协议的 FAT32 格式的储存设备，容量在 8G-32G 间，例如 U 盘，移动硬盘等。

推荐 U 盘品牌及型号如下表所示

品牌	型号	储存容量
aigo	U269	16G
Kingston	DTXM	32G
SanDisk	CZ73	32G
Lenovo	X3 Lite	32G



USB 存储器具有安全功能而在向驱动器存取时需要密码认证的产品，不能使用。

7.2 备份及加载对象

数据

- 用户程序文件 (program)
- 数据文件 (data)
- 系统设置文件 (system setting)
- TP 系统配置文件(TP config)
- 运控参数文件(MCCP)
- 日志(log)

软件

- 机器人应用软件
- 机器人运动控制软件



不包含操作系统和底层驱动。

备份方式选择：

1. 单个文件备份（文件管理界面）：选择选定的单个文件进行备份，保存到指定 u 盘路径的文件夹中。
2. 单一类型备份（文件管理界面）：选择某一文件类型，将此类型所有文件进行备份，保存到指定 u 盘路径的文件夹中。
3. 全部备份（文件管理界面）：备份所有文件，保存到指定 u 盘路径的文件夹中。
4. 镜像备份（文件管理界面）：镜像所列所有内容：所有文件、机器人应用软件、机器人运动控制软件，并以压缩包形式或其他形式，保存到指定 U 盘中。

加载方式选择：

1. 单个文件加载（文件管理界面）：选择路径选定单个文件进行加载。
2. 单一类型加载（文件管理界面）：选择路径，勾选某一文件类型，将此目录下该类型的所有文件进行加载。
3. 全部加载（文件管理界面）：选择 u 盘下的文件夹，将此目录下的所有可识别的机器人相关文件进行加载。
4. 镜像加载（文件管理界面）：选择 u 盘下文件夹中的镜像压缩包，选择镜像加载模式，解密解压压缩包，控制柜及 TP 中所有文件及安装的软件被替换（不涉及到操作系统）。该方式一般可用于恢复软件到某个版本，或者整体恢复某个已调试好的机器人工程应用。

7.3 其他功能描述

- 路径浏览：进行备份与恢复/加载时，提供路径浏览及选择功能，可以打开 U 盘中 Cabinet-USB\捷勃特 RobotBackup 路径下的文件夹，直至选择到需要操作的子目录为止。
- 文件识别：备份与恢复/加载时，只识别路径为 Cabinet-USB\捷勃特 RobotBackup 下的镜像文件（.zip）和普通的文件夹。保证此目录下只含有与机器人系统有关的文件。
- 文件夹编辑：允许在 Cabinet-USB\捷勃特 RobotBackup 目录下新建、删除和编辑命名文件夹。
- 覆盖提示功能：恢复/加载到控制柜时，如果有文件名相同的文件需确认是否覆盖。选项有“确认覆盖”、“跳过”、“取消备份/加载（之前已完成覆盖的文件无法复原）”，并具有勾选框“为之后的个文件执行相同的操作”。
- 进度确认：显示备份或加载的工作进度。

7.4 用户操作方式

备份操作：

1. 将 U 盘插入机器人控制柜上的 USB 接口处（示教器上的 USB 接口不支持备份操作）。
2. 将机器人切换为 SERVO_OFF 状态，依次点击“菜单按钮”→“管理”→“文件管理”→“备份”并打开文件管理界面，如下图 7.1 所示。



图 7.1 文件管理界面

3. 点击目标文件后面的正方形，选中目标文件后，点击“备份”按钮。

- 选择 U 盘路径和具体操作目录，如下图 7.2 和 7.3 所示，如有必要，建立或编辑文件夹。

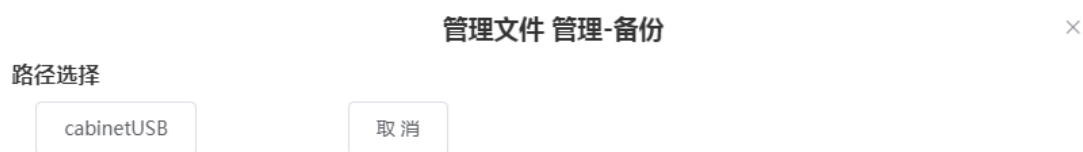


图 7.2 U 盘路径选择界面



图 7.3 文件目录界面

- 选择备份的目标文件夹，点击“开始备份”。
- 等待系统执行完成。

镜像操作：

- 将 U 盘插入机器人控制柜上的 USB 接口处（示教器上的 USB 接口不支持备份操作）。
- 将机器人切换为 SERVO_OFF 状态，依次点击“菜单按钮”→“管理”→“文件管理”→“备份”并打开文件管理界面。
- 点击“镜像”按钮。
- 选择 U 盘路径和具体操作目录，如有必要，建立或编辑文件夹。
- 选择备份的目标文件夹，点击“开始备份”。

6. 等待系统执行完成。

恢复/加载操作：

1. 插入 U 盘。
2. 将机器人切换到 SERVO_OFF 状态，依次点击“菜单按钮”→“管理”→“文件管理”→“恢复/加载”并打开文件管理界面。
3. 选择 U 盘路径和具体操作目录。
4. 选择需要加载的文件，点击“开始恢复”；等待系统执行完成。



图 7.4 恢复文件目录界面

注意事项

- 权限：在 admin 下可以进行该操作。
- 执行条件：在手动模式下，且机器人未进行任何运动指令（无论示教或程序），或者说机器人处于空闲状态下。
- 镜像恢复后，需要重启控制器与示教器。
- 在执行加载或者备份时，禁止进行任何其他软件操作。

8. 实用功能

8.1 程序偏移

对于程序中某一范围得动作语句，针对其中已经记录好位姿数据的运动语句，将这些位姿按照预先指定的规则进行统一变换，并将变换后的程序作为新程序或程序段。

偏移类型

- 一般偏移：平行或平行回转偏移
- 镜像偏移：相对所指定的对称面进行面对称偏移
- 圆周阵列偏移：在同一圆周上围绕回转轴进行角度偏移

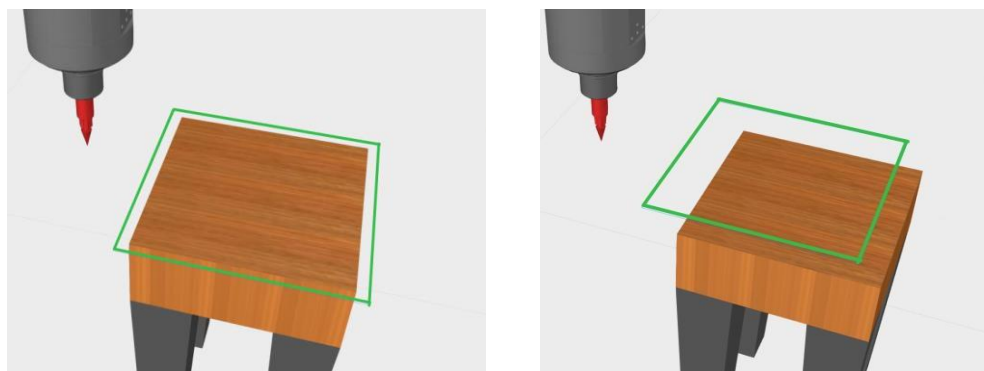


图 8.1 点位数据按一定规律偏移

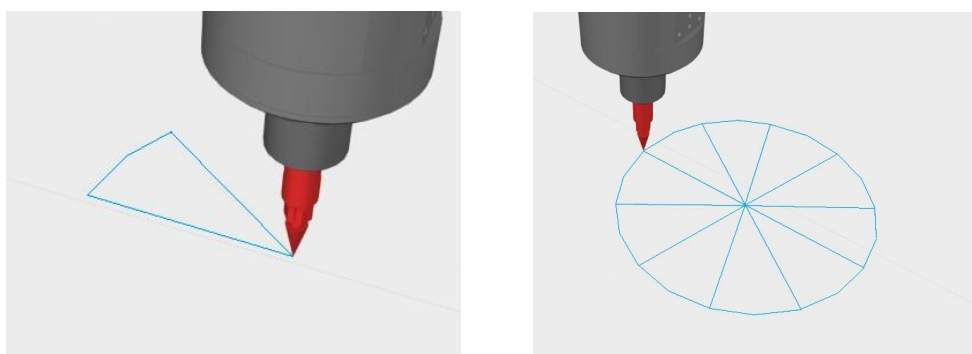


图 8.2 点位数据有规律性的重复

操作步骤：

依次点击“菜单按钮”→“应用”→“参考点”→“程序偏移”进入功能设置界面，如图 8.3 所示；

图 8.3 程序偏移设置界面

2.原始程序：在“原始程序”处选择需要偏移的程序名；若针对原程序中的某一范围进行偏移，则在“变换范围”处选择“局部”，并在“程序范围”处输入指定行号；

3.目标程序：在“目标程序”处输入新的程序名，偏移后的指令将生成在该新程序中；若输入现有的程序名，则需在“插入行”处指定插入到现有程序中的行号；

4.偏移方法设置：在“偏移类型”选择一般偏移、镜像偏移、圆周阵列偏移。

5.设置完成后，点击“执行变换”生成新建点。

6.点击“保存”生成新程序。

注意事项：

- 仅对 P[] 进行偏移计算，PR[] 不参与
- Cart 点偏移后仍为 Cart 点；Joint 点偏移后仍为 Joint 点
- Joint 点偏移后超出软限位，则该点偏移失败；复制的程序该点作为未示教记录的值

- Cart 点根据规则偏移，不作可达性判断

8.1.1 一般偏移

一般偏移

- 直接输入 X、Y、Z 偏移量
- 参考点：1 个原始点、1 个目标点

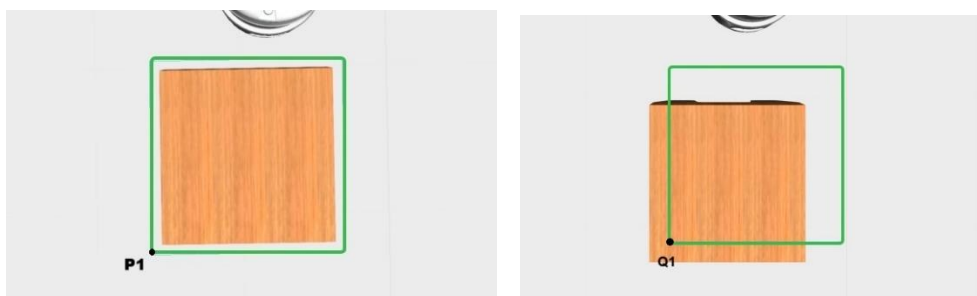


图 8.4 一般偏移

一般偏移带旋转

- 参考点：1 个原始点、1 个目标点

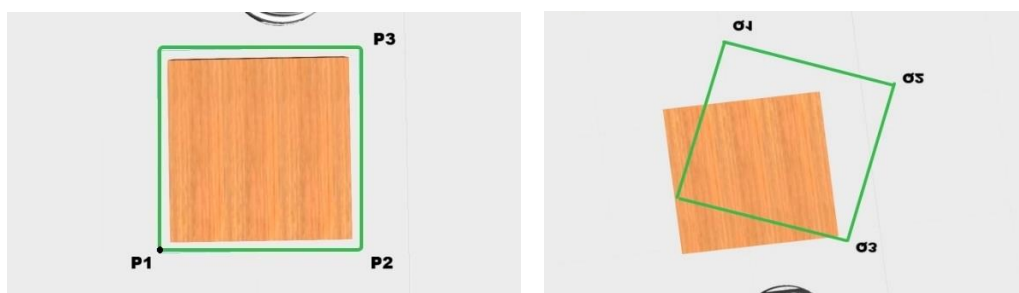


图 8.5 一般偏移带旋转

8.1.2 镜像偏移

镜像偏移

- 参考点：1 个原始点、1 个目标点

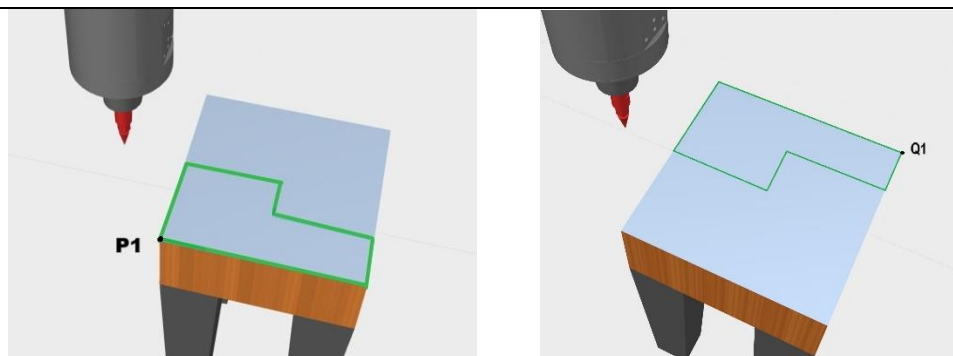


图 8.6 镜像偏移

镜像偏移带旋转

- 参考点: 3 个原始点、3 个目标点

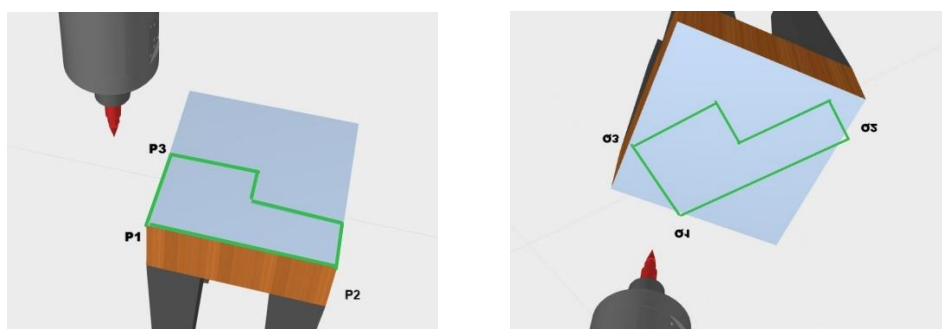


图 8.7 镜像偏移带旋转

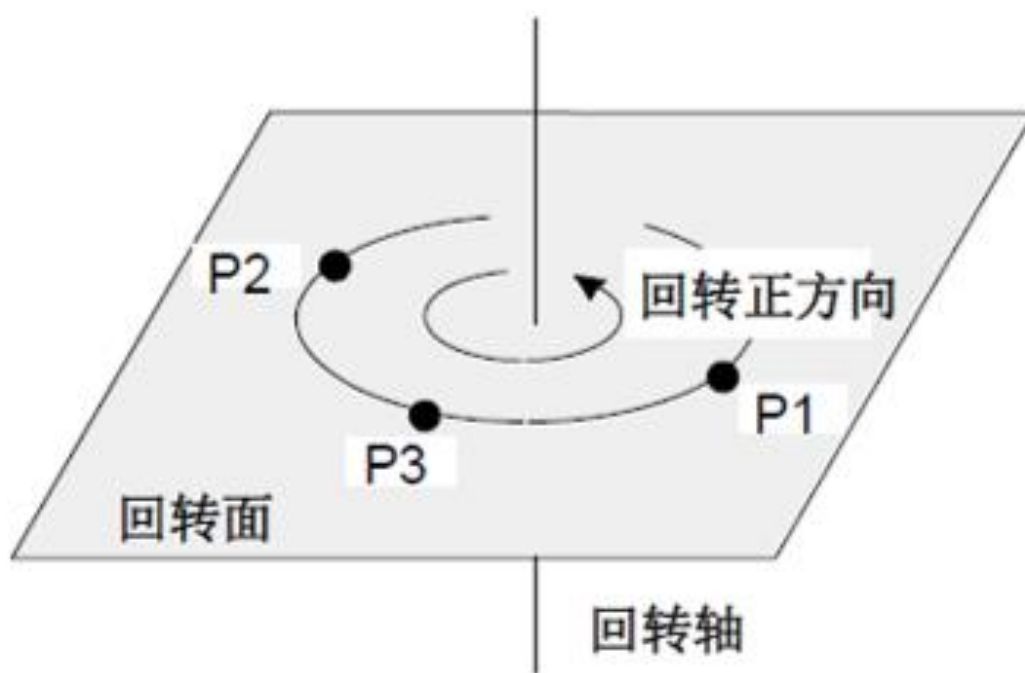
8.1.3 圆周阵列偏移

不指定旋转轴

- 参考点: 3 个

旋转面: 自动以 3 个点形成的旋转面

旋转轴: 3 个点形成圆的圆心与回转面的中垂线



无回转轴指定

图 8.8 不指定旋转轴

指定旋转轴

- 参考点：3 个参考点、1 个旋转轴上的点

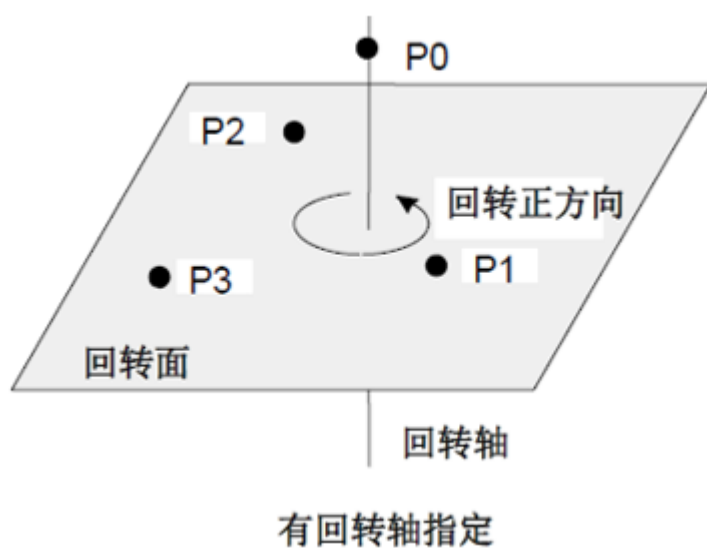


图 8.9 不指定旋转轴

8.2 基础空间防干涉

基础空间防干功能:当其它机器人或其它外围设备进入预先设定的干涉区域内时,即便向机器人发出进入干涉区域的移动指令,机器人也会自动停止,直到确认其它设备已经从干涉区域移走后,解除停止状态而自动重新开始动作。同时,可以设置对应输出,机器人实时判断 TCP 点位置,当 TCP 进入干涉区域时,会将指定 DO 状态置为 OFF,告知外围设备机器人已经进入干涉区域:当 TCP 离开干涉区域后,再将对应 DO 置 ON。

8.2.1 设置步骤

1.依次点击“菜单按钮”→“应用”→“参考点”→“基础空间防干涉”→“新建”创建进入配置界面,如图 8.10 所示。

基础空间防干涉功能			
#	名称	备注	关闭/激活

基本配置			
ID	<input type="text" value="1"/>	* 运动组	<input type="text" value="Group1"/>
* 名称	<input type="text"/>	备注	<input type="text"/>
* 输入信号	<input type="text" value="DI"/> <input type="text" value="0"/>	* 输出信号	<input type="text" value="DO"/> <input type="text" value="0"/>
* 监测优先级	<input type="text" value="高"/>	* 监测空间内/外	<input type="text" value="内部"/>

空间区域配置			
* UF	<input type="text" value="0"/>	* TF	<input type="text" value="0"/>
* 空间形状	<input type="text" value="立方体"/>	* 形状确定方法	<input type="text" value="顶点加边长法"/>
基准顶点	X: <input type="text" value="0"/> mm	Y: <input type="text" value="0"/> mm	Z: <input type="text" value="0"/> mm <input type="button" value="示教记录"/>
边长	X: <input type="text" value="0"/> mm	Y: <input type="text" value="0"/> mm	Z: <input type="text" value="0"/> mm

图 8.10 基础空间防干涉

2. 设置完成后→“保存”→“返回”→“关闭/激活”选择激活干涉区，如图 8.11 所示。

基本配置			
ID	<input type="text" value="1"/>	* 运动组	<input type="text" value="Group1"/>
* 名称	<input type="text" value="干涉区1"/>	备注	<input type="text"/>
* 输入信号	<input type="text" value="DI"/> <input type="text" value="DI[1]"/>	* 输出信号	<input type="text" value="DO"/> <input type="text" value="DO[2]"/>
* 监测优先级	<input type="text" value="高"/>	* 监测空间内/外	<input type="text" value="内部"/>

空间区域配置			
* UF	<input type="text" value="0"/>	* TF	<input type="text" value="0"/>
* 空间形状	<input type="text" value="立方体"/>	* 形状确定方法	<input type="text" value="顶点加斜对角点法"/>
基准顶点	X: <input type="text" value="-389.8643..."/> mm	Y: <input type="text" value="337.71444..."/> mm	Z: <input type="text" value="64.411748..."/> mm <input type="button" value="示教记录"/>
对角点	X: <input type="text" value="289.56784..."/> mm	Y: <input type="text" value="164.51728..."/> mm	Z: <input type="text" value="109.51567..."/> mm <input type="button" value="示教记录"/>

图 8.11 基础空间防干涉保存

8.2.2 基本配置界面

基本配置				
ID	1		* 运动组	Group1
* 名称			备注	
1 * 输入信号	DI	0	2 * 输出信号	DO 0
3 * 监测优先级	高		4 * 监测空间内/外	内部

图 8.12 基本配置界面

1.输入信号:

设定输入信号（该信号为自定义），将其它机器人或外部设备是否进入该干涉区域的状态进行输入。

输入信号断开时，机器人试图进入干涉区域内时，机器人进入保持状态。输入信号接通时，保持状态就被解除，系统自动地重新开始操作。



机器人从工具中心点进入干涉区域内的时刻起减速停止，所以机器人实际停止的位置，是进入干涉区域内的位置。

2.输出信号:

设定输出信号（该信号为自定义），工具中心点存在于干涉区域内时，该输出信号断开；存在于区域外时，该输出信号接通。

- 安全状态（干涉区域外）输出信号接通。
- 危险状态（干涉区域内）输出信号断开。

3.监测优先级:

在 2 台机器人上使用本功能的情况下，当 2 台机器人试图同时进入干涉区域内时，指定哪台机器人优先进入干涉区域。将本设定在“高”(优先考虑)侧的机器人先进入干涉区域在作业结束而从干涉区域退出后，设定在“低”(不优先考虑)侧的机器人进入干涉区域内。此外，2 台机器人必须设定为与对方一侧不同的设定。

注意：优先级有高和低两个选项，高优先级时，且检测在空间内部，则扩大检测范围（x、y、z 增加 10mm）；检测外部，则缩小检测范围（x、y、z 减少 10mm），低优先级则不会。

4. 监测空间（内部/外部）：

指定将设置的直方体的内部或者外部作为干涉区域。

8.2.3 空间区域配置界面



图 8.13 空间区域配置界面

空间区域配置内容	说明
UF	选择用户坐标系
TF	选择工具坐标系
空间形状	默认立方体
形状确定方法	顶点加边长法：从基准顶点到沿用户坐标系的 X、Y、Z 轴的直方体的边的长度(直方体各边必须平行于用户坐标系的坐标轴)。 顶点加斜对角点法：以基准顶点和对角顶点的直立方体为干涉区域。

顶点加边长法设置步骤：

1. 设置干涉直方体区域的顶点位置，移动机器人当前激活的 TCP 到想要设置为干涉区域顶点的位置，点击“示教记录”，则当前 TCP 点的位置就被设置为干涉直方体区域的一个顶点位置。也可以直接输入坐标值 X，Y，Z 的位置。

2. 在边长栏中指定从该顶点位置到沿当前用户坐标系的 X，Y，Z 轴的直方体的边的长度。



图 8.14 顶点加边长法设置

顶点加斜对角点法设置步骤：

1.移动机器人当前激活的 TCP 到想要设置为干涉区域**顶点**的位置,点击“示教记录”,则当前 TCP 点的位置就被设置为干涉长方体区域的一个顶点位置。也可以直接输入坐标值 X, Y, Z 的位置。



2.移动机器人当前激活的 TCP 到想要设置为干涉区域**斜对角点**的位置,点击“示教记录”。也可以直接输入坐标值 X, Y, Z 的位置。

图 8.15 顶点加斜对角点法设置

8.3 参考点

参考点是指预先设置好的一个或者若干个特定点位,启用该功能后,系统会实时判断机器人当前关节角度是否在设定参考位置的一定范围内(范围区间可设置),并且输出指定的信号,该功能通常用来告知外围设备机器人处于特定安全位置,或者机器人程序启动前是否在安全点位上。

可以设定 10 个参考位置。

进入设置界面：



依次点击“菜单按钮”→“应用”→“参考点”进入参考点设置界面,如图 8.16 所示。

图 8.16 参考点界面

具体步骤：

1. 点击“新建”创建参考点 RP。
2. 点击“编辑”进行参数设置。
3. 设定工具位于参考位置时输出的数字输出信号，可选择 DO 或 RO。注意避免与其它信号重复。
4. 要进行参考位置的示教有 2 种方法：示教记录当前位置、直接输入参考位置坐标值。在左侧输入坐标值,在右侧输入允许误差范围。请勿将浮动值为 0,基本上应设定在 0.1 以上。
5. 设定完成后按下激活键，点击保存修改。

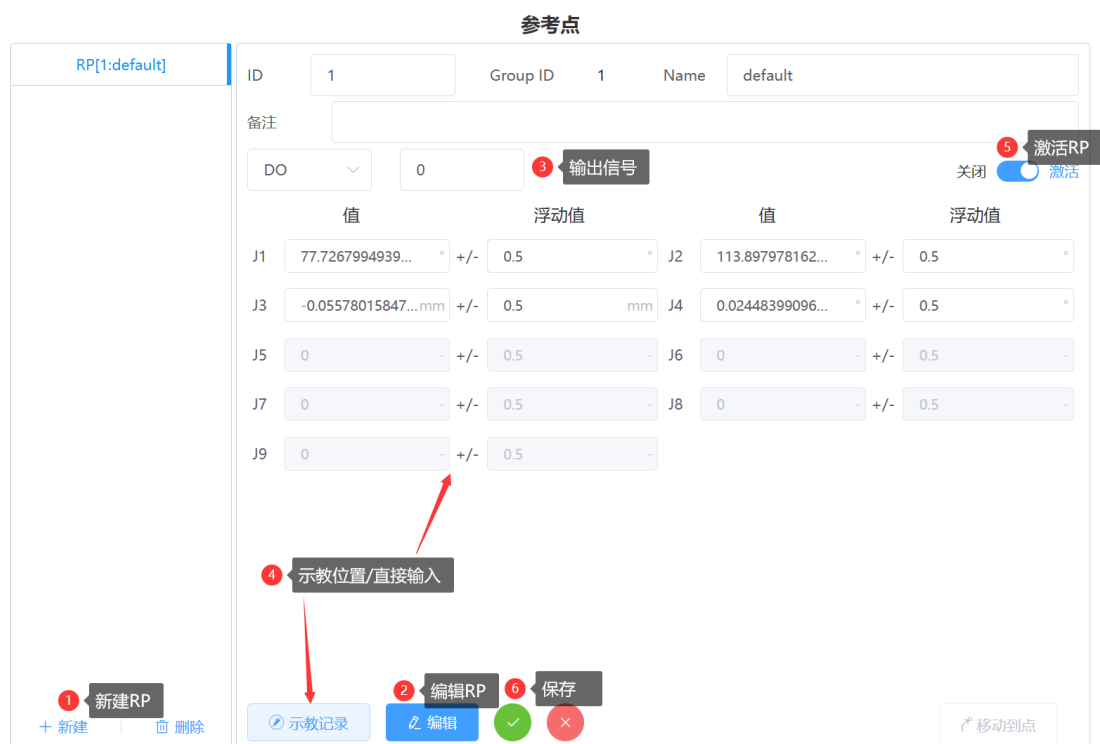


图 8.17 参考点设置界面

8.4 码垛拆垛

8.4.1 码垛功能描述

码垛，是将工件按照一定的顺序排列整齐地逐层进行摆放。

通过对一个工件的放置动作进行示教，指定接近退出点，拾取路径方式、排列方式、重叠方式，就能够简单地对全部工件的放置动作进行示教。另外，将已摆放工件按照相反顺序卸载的任务称为拆垛。

用户可以用最少的代码或者指令点选处理垛盘上的所有目标工件。

8.4.2 码垛结构和种类

码垛要实现多种多样的堆积效果，需要关键的路径样式和线路点信息。

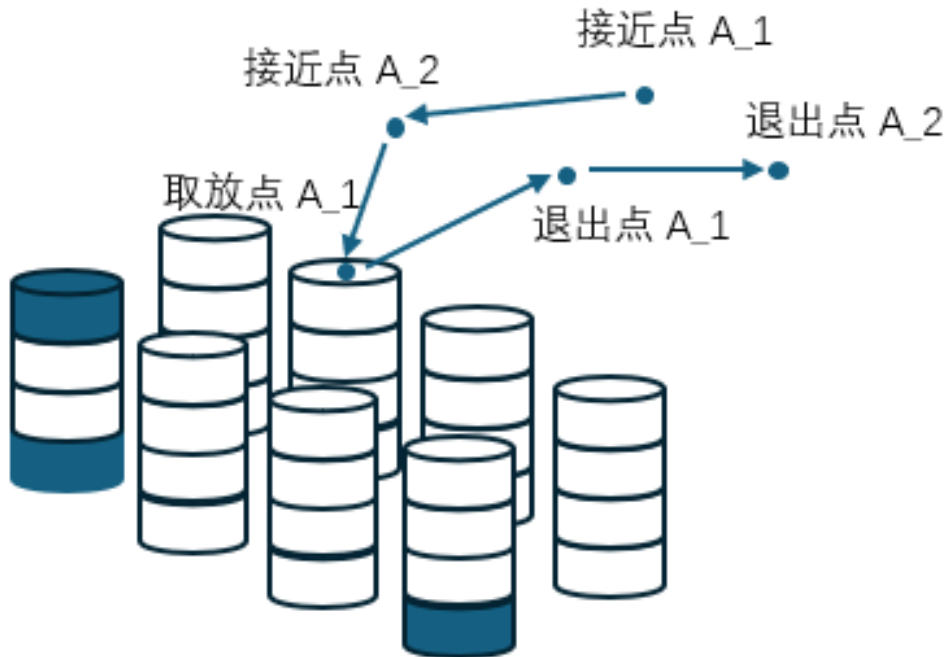


图 8.19 码垛路径样式示意图

码垛功能可实现工件姿势固定，均为[1,1,1]（码垛堆起始点）的姿态，底部形状为直线/平行四边形/梯形、有一条或多条进入和退出路线的码/拆垛类型。

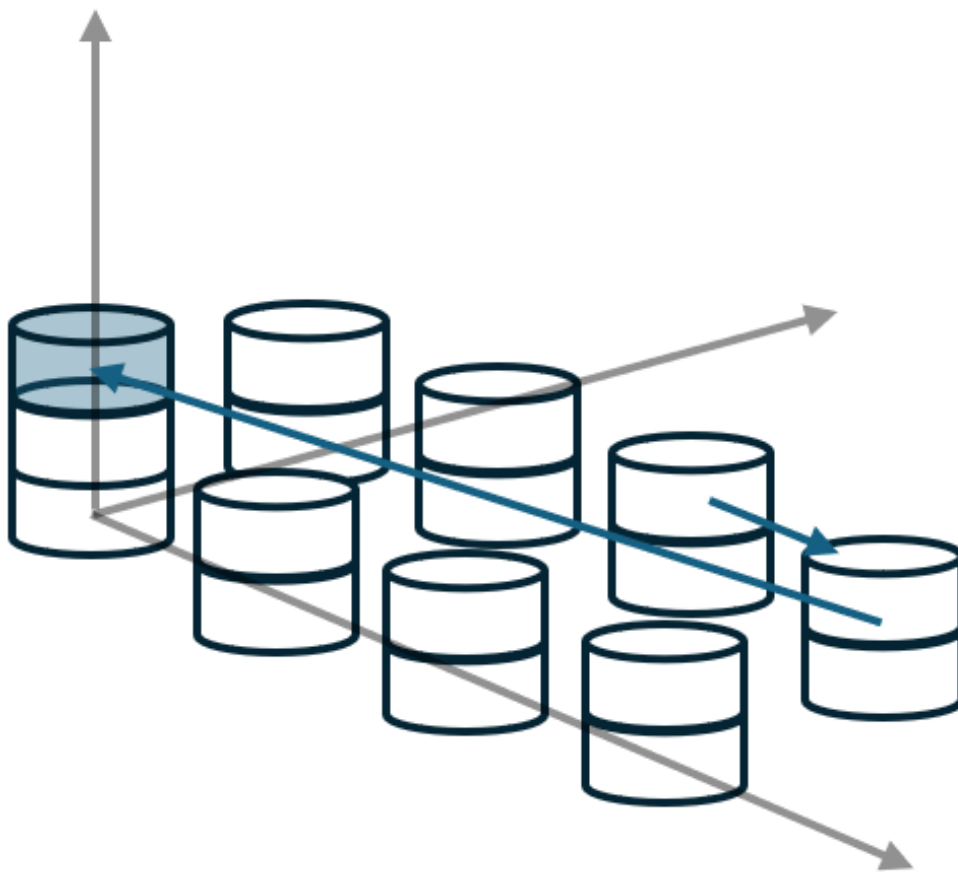


图 8.20 码垛形态和工件方向投影示意图

用户分别输入或示教垛堆码放的第一个工件所在的位置，以及行、列、层方向上的末位工件位置，并且填写垛堆行、列、层上的工件数量，软件根据这些参数生成填充点阵。

根据行、列、层的组合，一共 6 种组合方式，用于码/拆垛时的操作顺序，下图为

按行->列->层顺序的码放顺序图。垛本身的起始点[1,1,1]、行、列的方向不是固定的，是由示教的点位顺序控制的，第一个示教的点位为[1,1,1]，第二个点位确定行方向，第三个点位确定列方向；第四个点确定层的方向；



[3,2,2] → [4,2,2] → [1,1,3]

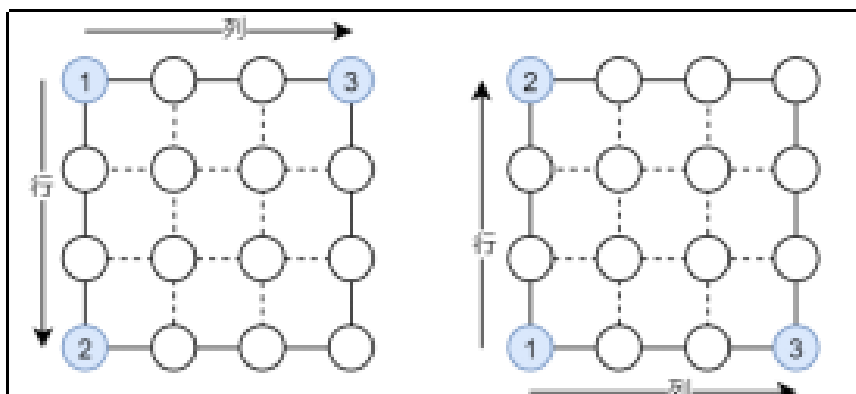


图 8.21 码垛顺序示意图

以 B 为例，在示教垛形态时，需要示教原点，行末尾，列末尾，最高点和辅助点来设定垛型，如下图为 4 行，2 列，4 层，无辅助点，需要示教点 [1,1,1],[1,2,1],[1,1,4],[4,1,1]。当垛形的行列层的示教点位存在重合时，后续重复的点无需示教。例如：行/列/层数 = [1,2,1]，需要示教的点：[1,1,1]、[1,2,1]；行/列/层数 =

[4,1,1], 需要示教的点: [1,1,1]、[4,1,1]; 例 4: 行/列/层数 = [4,2,1], 无辅助点, 需要示教的点: [1,1,1]、[4,1,1]、[1,2,1]; 当第一层的形状为梯形时, 可使用辅助点来辅助示教垛形态, 填充的点位按比例划分填充:

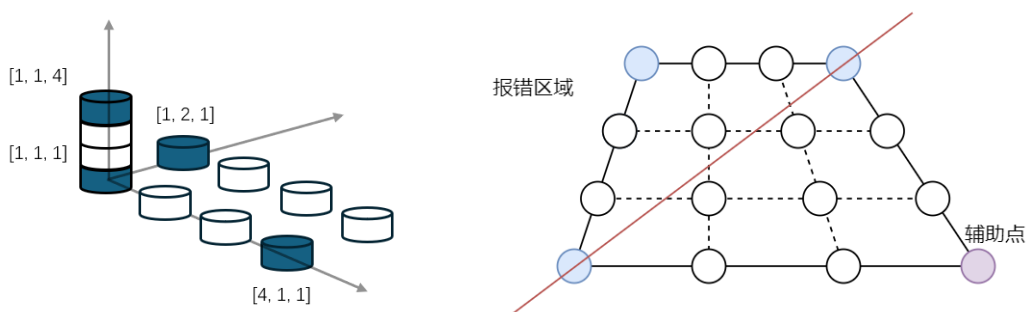


图 8.22 码垛辅助点示意图

注意:

辅助点应落在下图的红线右侧或下侧。

8.4.3 码垛方向和步进

码拆方向影响的是码/拆垛的起始位置以及行列层步进时的堆上点位值变化大小(加法或减法的固定值)及变化方向。

下表为码拆方向对码/拆垛的默认的初始化位置及下一步码拆方向的影响(不考虑人为干预的情况下)。

基础资料		初始值			下一步方向
垛姿态	方向	行	列	层	/
码垛	正序	1	1	1	行/列/层数值由小往大增长
	逆序	总行数	总列数	1	行/列数值由大往小减小
拆垛	正序	总行数	总列数	总层数	行/列/层数值由大往小减小
	逆序	1	1	总层数	行/列数值由小往大增长

步进值影响的是码/拆垛的行列层步进时的堆上点位值变化(加法或减法的固定

值)。当行/列/层方向上做加减步进值超出最大行/列/层的值时，认为当前行/列/层方向上的码/拆垛完成，可进行下一个行/列/层方向上的码/拆垛动作。

码/拆垛为行->列->层顺序的 2 行 2 列 2 层的垛，按照下列顺序进行码/拆垛：

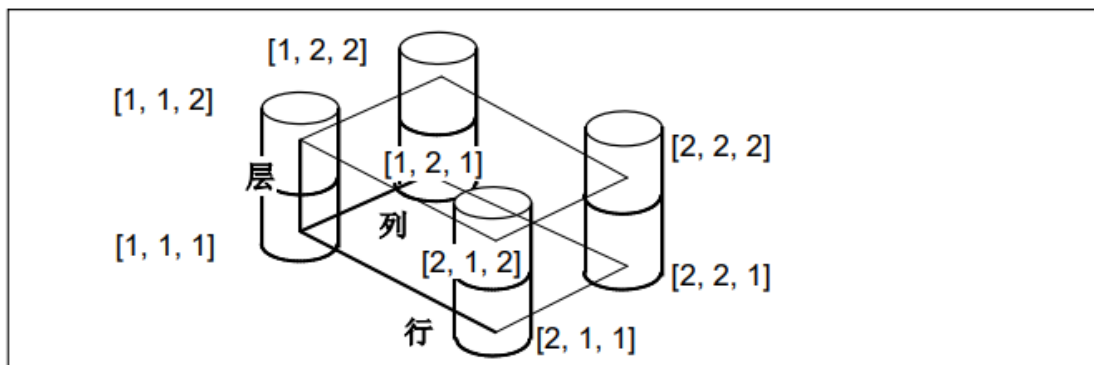


图 8.23 码垛辅助点示意图

2 行 2 列 2 层的码/拆垛操作顺序表：

码垛 (行->列->层)		拆垛 (行->列->层)	
正序，步进值：1	逆序，步进值：1	正序，步进值：1	逆序，步进值：1
[1,1,1]	[2,2,1]	[2,2,2]	[1,1,2]
[2,1,1]	[1,2,1]	[1,2,2]	[2,1,2]
[1,2,1]	[2,1,1]	[2,1,2]	[1,2,2]
[2,2,1]	[1,1,1]	[1,1,2]	[2,2,2]
[1,1,2]	[2,2,2]	[2,2,1]	[1,1,1]
[2,1,2]	[1,2,2]	[1,2,1]	[2,1,1]
[1,2,2]	[2,1,2]	[2,1,1]	[1,2,1]
[2,2,2]	[1,1,2]	[1,1,1]	[2,2,1]
[1,1,1]	[2,2,1]	[2,2,2]	[1,1,2]
...

4 行 2 列 2 层的码/拆垛操作顺序表:

码垛 (行->列->层)		拆垛 (列->行->层)	
正序, 步进值: 2	逆序, 步进值: 2	正序, 步进值: 3	逆序, 步进值: 3
[1,1,1]	[4,2,1]	[4,2,2]	[1,1,2]
[3,1,1]	[2,2,1]	[3,2,2]	[2,1,2]
[1,2,1]	[4,1,1]	[2,2,2]	[3,1,2]
[3,2,1]	[2,1,1]	[1,2,2]	[4,1,2]
[1,1,2]	[4,2,2]	[4,2,1]	[1,1,1]
[3,1,2]	[2,2,2]	[3,2,1]	[2,1,1]
[1,2,2]	[4,1,2]	[2,2,1]	[3,1,1]
[3,2,2]	[2,1,2]	[1,2,1]	[4,1,1]
[1,1,1]	[4,2,1]	[4,2,2]	[1,1,2]
...

8.4.4 码垛界面配置

使用码垛功能, 首先要配置码垛工艺工程, 进入路径“菜单-应用-码垛工艺”, 并按照码垛工艺工程的流程图完成配置, 如下图:



图 8.24 码垛工艺入口和配置流程示意图

8.4.4.1 码垛配置

手动模式（M、L）下且无用户程序驻留在内存中时（无用户程序处于暂停或运行状态），才可以进行码垛工艺工程的配置，否则只可查看，无法进行任何修改。

进入路径“菜单-应用-码垛”，点击“新建”会创建码垛工艺工程，如下图：



图 8.25 码垛工艺垛形和一般设置页面

点击新建可增加一个码垛工艺工程，最大支持 30 个工艺工程（序号 1-30），默认名称 default。

此时可点击“下一步”查看每个页面的参数，点击“编辑”后，可输入或修改当前参数。修

改后可通过“保存”按钮保存所有当前数据，或“撤销”按钮，丢弃所有数据，恢复编辑前状态

8.4.4.2 基础资料输入

基础资料对应 8.4.2 节和 8.4.3 中的码垛基本信息，修改基本信息会影响后续步骤中示教点的有效性和垛形。



图 8.26 码垛工艺基础资料设置页面

码拆类型 ()：类型是码垛，还是拆垛，默认码垛：

- 参数 1：码垛 (PALLET)；
- 参数 2：拆垛 (DEPALL)；

码拆方向 ()：用来确定码垛或者拆垛时的起始点位置及码拆方向：

- 参数 1：正序 ()
- 参数 2：逆序 ()

码拆顺序 ()：进行码垛和拆垛时的行(R)列(C)层(L)的顺序，排列组合共 6 种参数，默认行->列->层(R->C->L)：

- 参数 1：行->列->层(R->C->L)；
- 参数 2：行->层->列(R->L->C)；
- 参数 3：列->行->层(C->R->L)；
- 参数 4：列->层->行(C->L->R)；
- 参数 5：层->行->列(L->R->C)；

- 参数 6: 层->列->行(L->C->R);

8.4.4.3 示教垛形态

B 垛形的示教垛形态界面:



图 8.27 码垛工艺样式设置页面

示教垛形态 (1/1): B 型的码垛示教样式只有一种, 因此显示 (1/1) 即可, 垛形态示教点的姿态必须与 [1, 1, 1] 保持一致, 否则弹窗报错, 该点位示教无效。垛点 (PALLETIZING Points) 点位形式可参考用户坐标系的点位显示。

8.4.4.2 中辅助点为禁用时: 按顺序示教底部 3 个点和顶层的 1 个点位; 辅助点为启用时: 按顺序示教底部 4 个点和顶层的 1 个点位, 见下图:

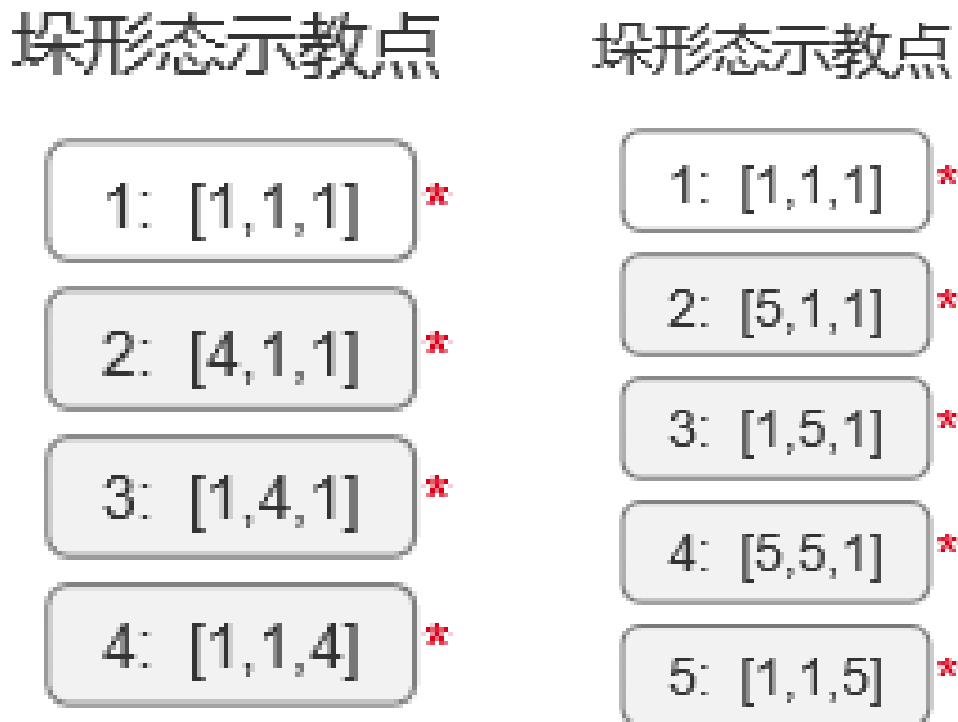


图 8.28 码垛工艺无辅助点和有辅助点的示教点列表

8.4.4.4 设定路径样式

需要给每个路径设定路径条件，用于搬运具体点位时的路径选择判断，并且要示教相应的路径点位；每一种路径样式的线路点可以有 0 至 8 个接近点或退出点。执行具体垛上点位码/拆时会调用匹配条件成功的路径样式。

按序新增一条路径样式，默认[*,*,*]，可填写三种类型参数，优先级：行=列=层，直接指定>余数指定>模糊匹配：

直接指定	1 至 255 整数	具体行/列/层的位置
余数指定法	m-n 整数	如 3-1 表示用垛上点位行/列/层值除以 3 余数为 1 则匹配成功
模糊匹配	*	表示垛上任一点位的行/列/层

点击红色按钮可进入示教页面，示教完成后将显示为蓝色按钮。



图 8.29 码垛工艺路径样式设置页面

根据接近点和退出点数量，路径样式编辑时会提供对应数量的接近点 A_i 和退出点 R_i ，所有点位需要全部写入完成后确认。



图 8.29 码垛工艺路径条件设置页面

8.4.4.5 示教路径点

未示教完成的路径样式为红色高亮，示教完成的为蓝色高亮；

同一个路径样式中，路径条件与示教的路径点没有关系，修改路径条件或示教路径互不影响；只有在用户程序运行过程中，根据匹配上的路径条件，再索引相关的路径点位；



图 8.30 码垛工艺路径样式设置页面

注意：

取放点仅有一个。在 8.4.4.4 中增多接近点和/或退出点，本节路径条件中原有点位数据保留，按序向下增加点位，所有点位变红；减少接近点和/或退出点，从大序号开始删除，未删除的原有点位数据保留，所有点位恢复红色 * 提示确认保存。

最后点击“完成”才会覆盖历史数据；否则配置的参数在离开当前 PALLETIZING[X:YYY]时根据弹窗的确认选项决定是否丢弃。

8.4.5 码垛程序

8.4.5.1 插入/编辑码垛寄存器

依次点击“插入指令”→“赋值及 IO”→“PL”，进入码垛寄存器页面，如下图。

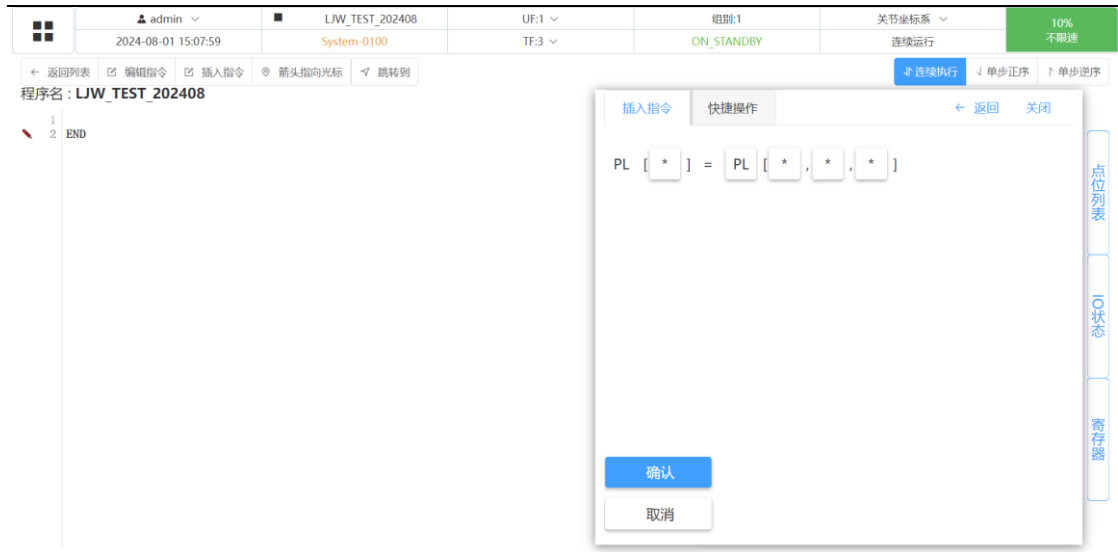


图 8.31 码垛工艺路径样式设置页面

PL[*] 只可选择码垛工艺工程建立完成的索引，码垛工艺工程未建立完成的，PL[*]索引不可见；赋值时，I、J、K 的值不可超出绑定的码垛工艺工程的行、列、层的数值配置；编辑码垛寄存器时，可参考 3.4.7 码垛寄存器中 PL[*]的插入及赋值规则。

8.4.5.2 插入码垛格式指令

依次点击“插入指令”→“码垛指令”→“PALLETIZING”，进入码垛指令选择页面，如下图所示。

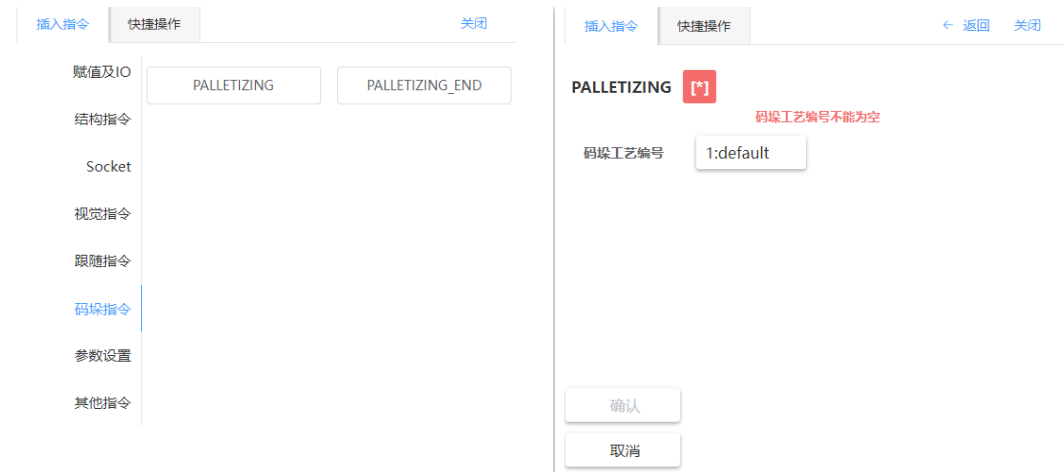


图 8.32 码垛工艺路径样式设置页面

PALLETIZING [*] 只可选择码垛工艺工程建立完成的索引，码垛工艺工程未建立完成的，PALLETIZING [*]索引不可见。

8.4.5.3 插入码垛动作指令

当选择的索引确定后，可通过依次点击“插入指令”→“码垛指令”→“PALLETIZING”，插入动作指令，默认动作指令格式为

MOVEJ PAL[x, xx], 30%, FINE:

如下图：



图 8.33 码垛工艺路径样式设置页面

微调码拆垛的工件姿态时，可通过使用码垛动作指令的 **OFFSET** 来实现，如若再配合路径条件，可进一步细化具体的某/行/列/层或具体 [I, J, K] 的工件姿态；例：

```
IF PL[1] = [1,1,1]
```

```
MOVEJ PAL[1, BTM], 500mm/s, FINE
```

```
ELSE IF PL[1] = [2,*,*]
```

```
MOVEJ PAL[1, BTM], 500mm/s, FINE, OFFSET PR[1]
```

依次点击“插入指令”→“码垛指令”→“PALLETIZING”，进入码垛指令选择页面，如下图，选择 **PALLETIZING_END** 指令。



图 8.34 码垛工艺路径样式设置页面

8.4.5.4 伪代码例程

#初始化

DO[1: 关闭真空] = OFF

MOVEL P[1: 初始位置], 500mm/s, FINE

//指定码垛开始的位置、码垛寄存器初始化

PL[1] = [1, 1, 1]

WHILE 1

#给料路径

MOVEL P[2: 给料位], 500mm/s, FINE

DO[1: 打开真空] = ON

MOVEL P[3: 安全移动位置], 500mm/s, FINE

#码垛指令开始

PALLETIZING [1: Test], B, LOAD, 3, 4, 5

#移动到安全码放位置

MOVEJ PAL[1, A_2], 30%, SD10

MOVEL PAL[1, A_1], 500mm/s, FINE

#码放

MOVEL PAL[1, BTM], 500mm/s, FINE

```
#松手爪  
DO[1: 关闭真空] = OFF  
#移动到安全退出位置  
MOVEL PAL[1, R_1], 500mm/s, SD10  
MOVEJ PAL[1, R_2], 50%, FINE  
#码垛结束, 码垛寄存器按照步进值自加  
PALLETIZING _END [1: Test ]  
#移动到安全返回位置  
MOVEJ P[4: 安全返回点], 500mm/s, FINE  
MOVEJ P[1: 初始位置], 500mm/s, FINE  
IF PL[1] = [3,4,5]  
#码垛满, 跳转  
GOTO LABEL[1:]  
END IF  
END WHILE  
LABEL[1:]  
END
```

8.4.6 注意事项

示教码垛工艺工程时, 不可有用户程序运行、暂停;

码垛格式指令、接近点/取放点/退出点动作指令、码垛结束指令 3 条指令存在于一个程序中才可发挥作用, 即使只将一个指令复制到子程序中, 也无法正常工作;

在包含码垛指令的程序中, 无法使用坐标系变换功能;

8.5 远程控制

远程控制适用于脱离示教器适用的场景, 上位机可以通过映射数字 IO 或总线 IO, 实现上使能、励磁、启动/恢复/暂停程序、修改全局速度、示教点动, 记录当前点位, 修改坐标系等功能。

在配置远程控制功能映射前, 应根据 2.1.1 节分配专用 IO 和 MH/MI 寄存器的地址, 如需使用总线和寄存器, 还应根据 5.1.6 节配置总线。

分配地址空间后，通过依次进入 菜单/应用/远程控制，可以进入具体功能的映射配置。

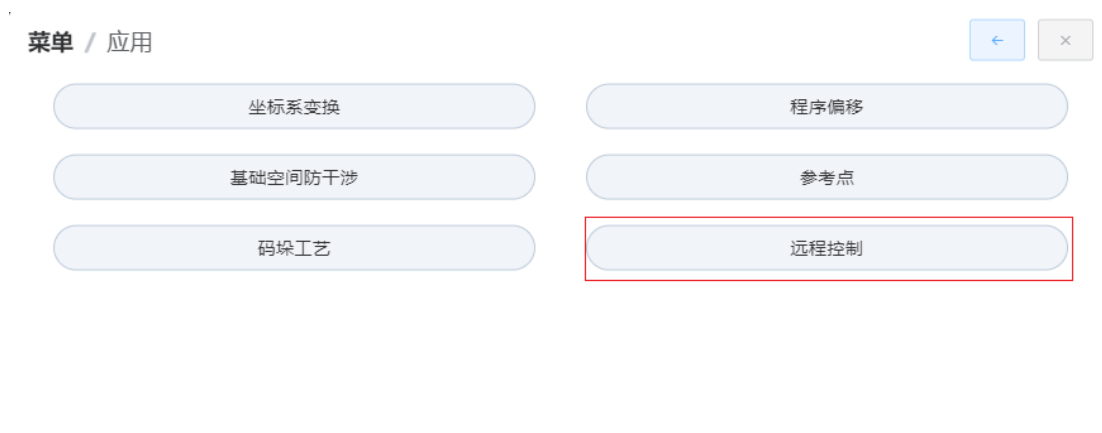


图 8.35 远程控制菜单入口

对于标准数字 IO 映射和现场总线 IO 映射的远程控制功能的配置如下，可以通过选项配置功能或数据映射的专用 IO (UI/UO) 或寄存器(MH/MI)。

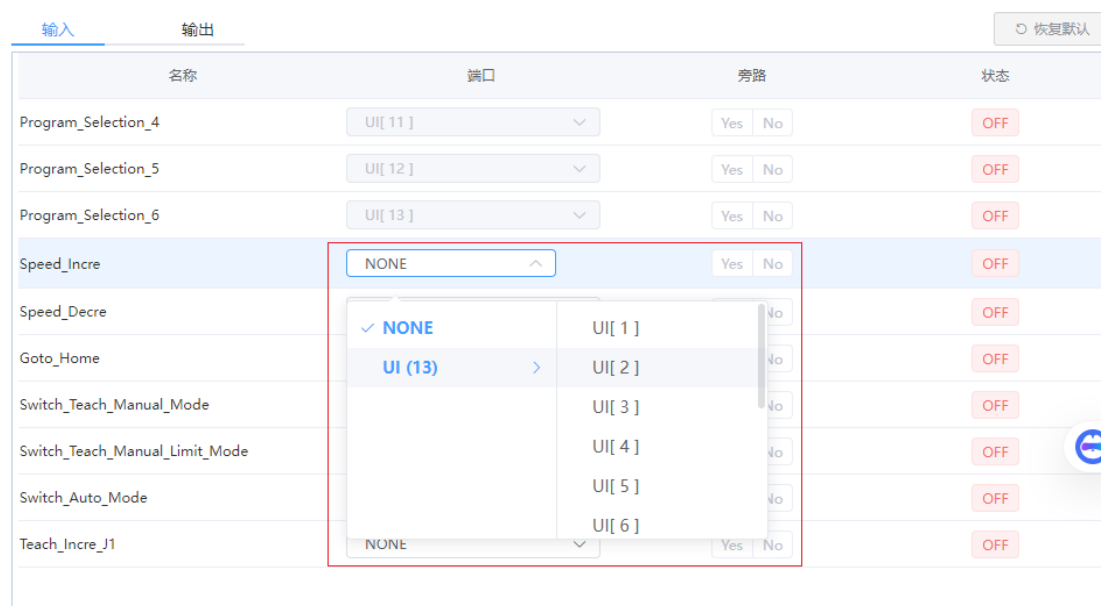


图 8.36 远程控制端口配置示意

8.5.1 信号类型

考虑到安全等因素，远程控制提供 6 种电平类型：

序号	信号类型	说明
1	低电平触发	给常态低电平即可触发相应操作
2	高电平触发	给常态高电平即可触发相应操作
3	上升沿触发	需要先给低电平，再给高电平触发相应操作，建议间隔 2ms 以上
4	下降沿触发	需要先给高电平，再给低电平触发相应操作，建议间隔 2ms 以上
5	特殊型	上升沿开始到下降沿的高电平触发模式，适用于手动示教场景。按下给高电平，松开给低电平
6	脉冲型	高/低电平周期性切换的触发模式，适用于检测设备在线状态等场景，防止设备假死

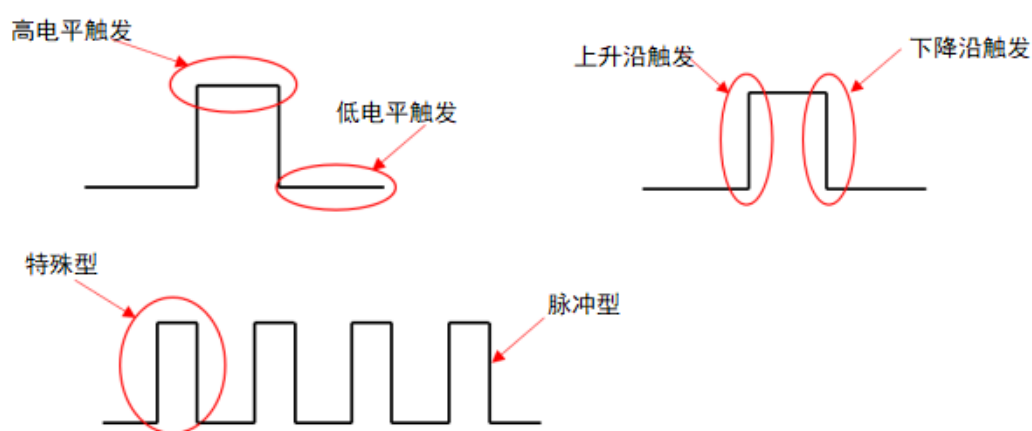


图 8.37 远程控制 IO 信号类型

8.5.2 输入地址映射表

序号	信号功能	信号类型	地址长度	备注

1	Servo Enable 伺服使能	特殊型	1	应用中，必须持续提供高电平
2	Pause Request 暂停	下降沿触发	1	常态下为高电平，机器人才可运行
3	Reset 报警复位	上升沿触发	1	需要在“伺服使能”为 ON 的前提下
4	Start&Restart 程序启动/恢复	下降沿触发	1	通过给一个脉冲信号产生下降沿
5	Abort Program 程序终止请求	下降沿触发	1	应用中，必须持续提供高电平和 UI2 一样，是事件信号，存在 OFF 时，多次触发信号导致系统卡死的风险。
6	Selection Strobe 触发	高电平触发	1	用于记录程序号，系统在触发信号的高电平持续期间内，读取 Program Selection1-6 的信号。
7	主程序号启动 MPLCS Start	下降沿触发	1	用于执行程序，ON 状态应保持在 100ms 以上，然后转变为 OFF 时，系统识别到下降沿，程序启动。
8	程序序号 Program Selection 1	高电平触发	1	
9	程序序号 Program Selection 2	高电平触发	1	
10	程序序号 Program Selection 3	高电平触发	1	

11	程序序号 Program Selection 4	高电平触发	1	
12	程序序号 Program Selection 5	高电平触发	1	
13	程序序号 Program Selection 6	高电平触发	1	
14	速度加 V+	上升沿触发	1	
15	速度减 V-	上升沿触发	1	
16	原点回归	特殊型	1	需要提供 home 点的 ID，执行该操作后，当前直接运动方式会变为 MovJ
17	切换到 手动示教模式	上升沿触发	1	
18	切换到 手动限速示教模式	上升沿触发	1	
19	切换到 自动模式	上升沿触发	1	
20	示教 J1/X +	特殊型	1	
21	示教 J2/Y +	特殊型	1	
22	示教 J3/Z +	特殊型	1	
23	示教 J4/A +	特殊型	1	

24	示教 J5/B+	特殊型	1	
25	示教 J6/C+	特殊型	1	
26	示教 J1/X -	特殊型	1	
27	示教 J2/Y -	特殊型	1	
28	示教 J3/Z -	特殊型	1	
29	示教 J4/A -	特殊型	1	
30	示教 J5/B-	特殊型	1	
31	示教 J6/C-	特殊型	1	
32	把当前位置写入当前 PR 变量	特殊型	1	
33	把输入点位写入当前 PR 变量	特殊型	1	
34	直接运动到当前 PR 变量位置	特殊型	1	
35	PR 变量读取成功	上升沿触发	1	
36	总线通讯心跳信号	脉冲信号	1	以判断双方在线和切换到 PC 模式。输入心跳未配置或者输入心跳脉冲 5s 内没有发生变化均认为离线
37	Home 点的 ID		16	此项及以下各项应选择 IO 组或 MH 寄存器映射

38	当前 PR 变量的 ID		16	
39	PR 变量写入的臂参数 1		16	4 轴机器人 J2 关节配置 1: 手臂在右 -1: 手臂在左
40	PR 变量写入的臂参数 2		16	6 轴机器人 J1 关节配置 1: 手臂在前 -1: 手臂在后
41	PR 变量写入的臂参数 3		16	6 轴机器人 J3 关节配置 1: 手臂在上 (肘在腕到 2 轴连线上方) -1: 手臂在下 (肘在腕 2 轴连线下方)
42	PR 变量写入的臂参数 4		16	6 轴机器人 J5 关节配置 1: 腕向下翻转 -1: 腕向上翻转
43	PR 变量写入的回转参数 1		16	1 轴回转数
44	PR 变量写入的回转参数 2		16	2 轴回转数
45	PR 变量写入的回转参数 3		16	3 轴回转数
46	PR 变量写入的回转参数 4		16	4 轴回转数
47	PR 变量写入		16	5 轴回转数

	的回转参数 5			
48	PR 变量写入的回转参数 6		16	6 轴回转数
49	PR 变量写入的坐标系		16	0 选择关节坐标系 1 选择笛卡尔坐标系
50	PR 变量写入的 J1/X 坐标		32	
51	PR 变量写入的 J2/Y 坐标		32	
52	PR 变量写入的 J3/Z 坐标		32	
53	PR 变量写入的 J4/A 坐标		32	
54	PR 变量写入的 J5/B 坐标		32	
55	PR 变量写入的 J6/C 坐标		32	
56	坐标系选择		16	修改坐标系会影响当前位置输出的显示，即同一位置在不同的坐标系下有不同的表现形式。 0 关节坐标系 1 基座坐标系 2 世界坐标系 3 用户坐标系 4 工具坐标系 5 RTCP 用户坐标系

6 RTCP 工具坐标系				
57	工具坐标系号 选择		16	修改工具坐标系，会影响：当前位置按笛卡尔坐标系写入 PR 的结果，点动到 PR 点的最终位置和运动轨迹和输出结果
58	用户坐标系号 选择		16	修改用户坐标系，会影响：当前位置按笛卡尔坐标系写入 PR 的结果，点动到 PR 点的最终位置和运动轨迹和输出结果
59	全局速度设置		16	1-10,000 表示 0.01% - 100%

8.5.3 输出地址映射表

序号	信号功能	信号类型	地址长度	备注
1	CMDENBLE 允许外围设备 控制机器人的 状态信号		1	下列条件成立时输出高电平： 1)机器人运行状态为“On-Standby” 2)处于“Auto”模式 3)程序执行模式没有选择“单步执行”或者“逆序执行”。(该信号高电平时，代表允许使用《自动模式下程序的启动方式》来启动序,或者暂停恢复。具体根据当前是否处于 Paused)
2	“暂停中”状态 信号 Paused		1	程序执行状态处于“暂停状态 Paused”时，该信号为 ON（即机器人暂停）。
3	FAULT		1	报警信号在系统中发生报警时产生

	报警信号			<p>输出，可以通过 RESET 来进行复位。</p> <p>注：当系统发出警告类报警时 (Warning)，该信号不输出。</p>
4	Program Running 程序正在运行中信号		1	<p>为 ON 时，表明正在运行程序中，即程序执行状态为“Executing”。</p>
5	伺服状态信号 Servo Status		1	<p>机器人运行状态为“Working”、“On-Standby”、“Servo-ON”时，该信号置高电平。</p> <p>“Servo-OFF”时置低电平。</p>
6	程序选择确认 Selection Check Request		1	<p>仅在“自动模式程序启动方式”设置为“MPLCS”时有效。</p>
7	主程序号启动完成 MPLCS Start Done		1	<p>仅在“自动模式程序启动方式”设置为“MPLCS”时有效。</p>
8	程序 1 选择确认 Selection Confirm 1		1	<p>仅在“自动模式程序启动方式”设置为“MPLCS”时有效。</p> <p>在接收到 Selection Strobe 信号后，机器人控制器会去读 UI[5]-UI[10]的状态，并将其反馈给上位确认。</p>
9	程序 2 选择确认 Selection Confirm 2		1	

10	程序 3 选择确认 Selection Confirm 3		1	
11	程序 4 选择确认 Selection Confirm 4		1	
12	程序 5 选择确认 Selection Confirm 5		1	
13	程序 6 选择确认 Selection Confirm 6		1	
14	总线通讯心跳信号		1	脉冲信号
15	PR 变量写入是否成功		1	输出`ON`表示 PR 变量写入成功, `OFF`表示 PR 变量写入失败; 未触发写入前, 维持上一次的状态
16	手动示教模式输出		1	
17	手动限速示教模式输出		1	
18	自动模式输出		1	
19	机器人正在运动中信号		1	机器人在运动过程中输出 ON, 到达指定位置输出 OFF

20	PR 变量读取的臂参数 1	IO 组或 MI 寄存器	16	
21	PR 变量读取的臂参数 2	IO 组或 MI 寄存器	16	
22	PR 变量读取的臂参数 3	IO 组或 MI 寄存器	16	
23	PR 变量读取的臂参数 4	IO 组或 MI 寄存器	16	
24	PR 变量读取的回转参数 1	IO 组或 MI 寄存器	16	
25	PR 变量读取的回转参数 2	IO 组或 MI 寄存器	16	
26	PR 变量读取的回转参数 3	IO 组或 MI 寄存器	16	
27	PR 变量读取的回转参数 4	IO 组或 MI 寄存器	16	
28	PR 变量读取的回转参数 5	IO 组或 MI 寄存器	16	
29	PR 变量读取	IO 组或	16	

	的回转参数 6	MI 寄存器		
30	PR 变量读取的坐标系	IO 组或 MI 寄存器	16	
31	PR 变量读取的 J1/X 坐标	IO 组或 MI 寄存器	32	
32	PR 变量读取的 J2/Y 坐标	IO 组或 MI 寄存器	32	
33	PR 变量读取的 J3/Z 坐标	IO 组或 MI 寄存器	32	
34	PR 变量读取的 J4/A 坐标	IO 组或 MI 寄存器	32	
35	PR 变量读取的 J5/B 坐标	IO 组或 MI 寄存器	32	
36	PR 变量读取的 J6/C 坐标	IO 组或 MI 寄存器	32	
37	当前位置 J1/X 坐标	IO 组或 MI 寄存器	32	
38	当前位置 J2/Y 坐标	IO 组或 MI 寄存	32	

		器		
39	当前位置 J3/Z 坐标	IO 组或 MI 寄存器	32	
40	当前位置 J4/A 坐标	IO 组或 MI 寄存器	32	
41	当前位置 J5/B 坐标	IO 组或 MI 寄存器	32	
42	当前位置 J6/C 坐标	IO 组或 MI 寄存器	32	
43	当前全局速度	IO 组或 MI 寄存器	16	1-10,000 表示 0.01% - 100%
44	当前坐标系	IO 组或 MI 寄存器	16	
45	当前工具坐标系号	IO 组或 MI 寄存器	16	
46	当前用户坐标系号	IO 组或 MI 寄存器	16	

注意：主程序号启动 MPLCS 相关信息详见《自动模式下程序的启动方式》中“MPLCS”的流程描述。

9. 报警信息列表

9.1 报警事件说明

捷勃特支持以下报警事件等级划分以及对应的行为策略如图 9.1 所示：

报警等级对应的行为和安全策略					
事件等级	机器人运动	伺服母线电源	用户程序	范围	报警对应的安全停止策略
INFO	不受影响	不受影响	不受影响	...	无
PAUSE.L PAUSE.G	减速后停止	不断开（松闸）	暂停	局部	无
STOP.L STOP.G	减速后停止	断开		整体	停止类型：1类停止
SERVO1	瞬时停止			整体	停止类型：0类停止
ABORT.L ABORT.G	减速后停止	断开(抱闸)	终止	局部	停止类型：1类停止
SERVO2	瞬时停止	断开（抱闸）		整体	停止类型：0类停止
SYSTEM				整体	停止类型：0类停止 与系统相关的重大问题，此报警发生后应当禁止所有的机器人操作，解决问题后需要关机重启，初始化系统。
Warning	不受影响	不受影响	不受影响	...	无

图 9.1 事件等级与策略

当前报警事件说明：

“Info”以上事件等级的事件（不含 Info），被定义为报警。

“当前报警”指的是现在还没有成功复位，报警继续有效，它的“事件等级”对应的处理行为仍会继续对系统产生限制或警告的报警。与之对应，如果报警被复位了，就不再是当前有效的报警，而是过去发生过的报警了。则这样的报警就会从“当前报警”列表中消失，而只存在于“事件历史”中。

当前报警会显示在 UI 状态栏的“事件信息”区域，用于提示用户当前系统中是否存在仍起效的报警，方便用户进行故障排查和调试机器人。

注意以下几点：

- 只有当前最高等级的事件，会被显示在事件信息栏。
- “Info”等级及更高事件等级的事件都会被显示在“事件信息”状态栏。
- 当前报警也有独立的显示界面用于显示所有当前报警，但是在该界面中“Info”等级的事件是不显示的（因为 Info 实际不是报警）。

9.1.1 相关界面及功能描述



图 9.2 状态栏事件信息区域

如图 9.2 所示，为报警事件状态栏。用户可以点击此状态栏上的事件信息区域，进入当前报警界面，如图 9.3 所示。



图 9.3 当前报警界面

当前报警界面用于显示系统中目前存在的所有报警。按照时间顺序从当前往过去进行依次排序。用户点击任意一个报警，进入其详细说明界面。



图 9.4 报警详细界面

如图 8.4 所示，为当前报警-单个报警的详细界面。此界面是针对某一个报警的具体详细描述，包含了报警代码、报警等级、报警名称、报警的描述、造成的后果、产生的原因、解决的方案。。用户可以通过翻页进行不同报警之间的切换。切换的列表为当前报警界面内拥有的报警。

当 **Reset** 信号给控制器进行复位操作。当控制器完成复位操作后（无论复位是成功还是失败），“当前报警”界面刷新。如果复位后存在报警，则按报警的时间顺序显示到当前报警界面中；若没有，则为空页面，示教器界面状态栏上不再做任何报警显示。

用户还可以从当前报警界面点击历史事件按钮，进入历史事件界面。

9.2 历史事件

概述

历史事件是记录用户对机器人系统的操作，以及系统内部为之而发生的报警、提示、状态跳转等；例如，用户登录、报警、暂停操作、电机上电、接触器松开、机器

人进入手动限速模式、复位、标定等等一系列事件。



Event Log 存储大小暂定为 250M，超过后新 Log 会覆盖离现在时间最久远的 Log 信息。

9.2.1 相关界面及功能描述

报警编码	描述	发生时间
1	System-2196 TpComm模块日志run /rpc/tp_comm/getPublishTable success*	2023-02-17 18:44:33
2	Operation-0065 运行模式切换至SlowlyManual	2023-02-17 18:44:33
3	System-0069 worker工作线程启动*	2023-02-17 18:44:33
4	System-2196 TpComm模块日志run /rpc/tp_comm/getRpcTable success*	2023-02-17 18:44:32
5	Operation-0018 admin登出成功	2023-02-17 18:19:12
6	Operation-0009 屏幕锁定	2023-02-17 18:12:51
7	Operation-0009 屏幕锁定	2023-02-17 17:59:12
8	Safety-4005 安全板错误总标志位*	2023-02-17 17:49:41
9	Safety-2306 SafetyBoardDebug模块故障代码0x2000*	2023-02-17 17:49:40
10	Safety-4006 安全板限速标志位*	2023-02-17 17:49:40

图 9.5 事件日志/历史界面

依次点击“菜单按钮”→“快捷菜单”→“历史事件”进入图 8.5 所示，为历史事件日志界面。该界面按时间顺序从现在到过去依次排列出所有发生过的事件。

用户可以通过筛选器进行筛选；筛选有两类，一个是按报警类别或者等级进行划分用户选择后显示内容为用户所选择的类别相关的；另一个滤波器为时间滤波器，用户可以选择仅显示某段时间内的事件日志。

如果用户点击事件日志中的一个事件记录，也会自动进入其详细说明界面。

10. SOCKET 错误代码表

本章介绍 SOCKET 指令（见 3.8.12 节）中可选参数 parameter 值的含义

Parameter 值	含义	中文释义
0	Success	成功
1	Operation not permitted	操作不允许
2	No such file or directory	文件/路径不存在
3	No such process	进程不存在
4	Interrupted system call	系统调用被中断
5	Input/output error	I/O 错误
6	No such device or address	设备/地址不存在
7	Argument list too long	参数列表过长
8	Exec format error	执行格式错误
9	Bad file descriptor	文件描述符错误
10	No child processes	子进程不存在
11	Try again	暂时无法完成操作，需要再次尝试
12	Out of memory	内存不足
13	Permission denied	权限错误
14	Bad address	地址错误
15	Block device required	块设备请求

16	Device or resource busy	设备或资源忙
17	File exists	文件已存在，即试图创建已经存在的文件或目录时发生错误
18	Cross-device link	无效的交叉链路
19	No such device	设备不存在
20	Not a directory	目录不存在
21	Is a directory	试图对一个目录执行不允许的操作
22	Invalid argument	参数无效
23	File table overflow	系统打开文件的数量超过了操作系统允许的限制
24	Too many open files	打开的文件过多
25	Not a typewriter	操作不适用于给定的对象
26	Text file busy	文本文件忙
27	File too large	文件过大
28	No space left on device	设备上没有剩余空间
29	Illegal seek	非法搜索
30	Read-only file system	只读文件系统
31	Too many links	链接太多
32	Broken pipe	管道断开

33	Math argument out of domain of func	数学域错误，指定的参数超出了函数定义的有效范围
34	Math result not representable	数值结果超出函数可表示范围
35	Resource deadlock would occur	死锁状态，发生多个进程或线程相互等待资源的情况
36	File name too long	文件名过长
37	No record locks available	无锁可用
38	Invalid system call number	系统调用或操作不被支持或未实现
39	Directory not empty	目录非空
40	Too many symbolic links encountered	符号链接循环或嵌套层级过多
41	Operation would block	操作将会阻塞，但当前设置为非阻塞模式
42	No message of desired type	没有所需类型的消息
43	Identifier removed	标识符已删除
44	Channel number out of range	通道号超出范围
45	Level 2 not synchronized	2级不同步
46	Level 3 halted	3级停止
47	Level 3 reset	3级复位
48	Link number out of	链接号超出范围

	range	
49	Protocol driver not attached	协议驱动程序没有连接
50	No CSI structure available	没有可用的 CSI 结构
51	Level 2 halted	2 级停止
52	Invalid exchange	交换无效
53	Invalid request descriptor	请求描述符无效
54	Exchange full	资源不足
55	No anode	无阳极
56	Invalid request code	无效请求码
57	Invalid slot	无效槽位
58	Resource deadlock would occur	死锁状态，发生多个进程或线程相互等待资源的情况，同 35
59	Bad font file format	错误的字体文件格式
60	Device not a stream	设备不是字符流
61	No data available	没有可用的数据
62	Timer expired	定时器超时
63	Out of streams resources	流资源不足
64	Machine is not on the network	机器不在网络上

65	Package not installed	包未安装
66	Object is remote	远程的对象
67	Link has been severed	链接被断开
68	Advertise error	广告错误
69	Srmount error	Srmount 错误，操作被拒绝，因为文件系统是只读的
70	Communication error on send	发送过程中通讯错误
71	Protocol error	协议错误
72	Multihop attempted	多跳尝试
73	RFS specific error	RFS 特定错误
74	Not a data message	错误的消息
75	Value too large for defined data type	值超出指定的数据类型范围
76	Name not unique on network	网络上名称不唯一
77	File descriptor in bad state	文件描述符状态错误
78	Remote address changed	远程地址被改变
79	Can not access a needed shared library	无法访问需要的共享库
80	Accessing a corrupted shared library	访问损坏的共享库

81	.lib section in a.out corrupted	a.out 中有 .lib 段被损坏
82	Attempting to link in too many shared libraries	试图链接太多的共享库
83	Cannot exec a shared library directly	不能直接运行共享库
84	Illegal byte sequence	非法的或无效的字节序列
85	Interrupted system call should be restarted	应重新启动被中断的系统调用
86	Streams pipe error	流管道错误
87	Too many users	用户太多
88	Socket operation on non-socket	在非套接字上进行套接字操作
89	Destination address required	需要目标地址
90	Message too long	消息太长
91	Protocol wrong type for socket	协议类型错误
92	Protocol not available	协议不可用
93	Protocol not supported	协议不支持
94	Socket type not supported	不支持套接字类型
95	Operation not supported on	操作不被支持或不可用

	transport endpoint	
96	Protocol family not supported	不支持协议族
97	Address family not supported by protocol	协议不支持地址族
98	Address already in use	地址已被使用
99	Cannot assign requested address	无法分配请求的地址
100	Network is down	网络中断
101	Network is unreachable	网络不通
102	Network dropped connection because of reset	网络在重置时断开连接
103	Software caused connection abort	软件导致连接终止
104	Connection reset by peer	连接被对方复位
105	No buffer space available	没有可用的缓存空间
106	Transport endpoint is already connected	传输端点已连接
107	Transport endpoint is not connected	传输端点未连接
108	Cannot send after transport endpoint shutdown	传输端点关闭后不能再发送

109	Too many references: cannot splice	引用过多：无法接合
110	Connection timed out	连接超时
111	Connection refused	连接被拒绝
112	Host is down	主机宕机
113	No route to host	无法路由到主机
114	Operation already in progress	操作已在进程中
115	Operation now in progress	操作正在进行
116	Stale file handle	失效的文件句柄
117	Structure needs cleaning	结构需要清理
118	Not a XENIX named type file	不是 XENIX 命名类型的文件
119	No XENIX semaphores available	没有可用的 XENIX 信号量
120	Is a named type file	是命名类型的文件
121	Remote I/O error	远程 I/O 错误
122	Quota exceeded	超出磁盘配额
123	No medium found	没有发现介质
124	Wrong medium type	错误的介质类型
125	Operation Canceled	取消操作

126	Required key not available	Key 不可用
127	Key has expired	Key 已过期
128	Key has been revoked	Key 被撤销
129	Key was rejected by service	Key 被拒绝服务
130	Owner died	所有者死亡
131	State not recoverable	状态不可恢复
132	Operation not possible due to RF-kill	由于 RF-kill 而无法操作
133	Memory page has hardware error	硬件故障导致内存中的数据出现错误
999	Recv timeout	接受超时

联系我们



上海捷勃特机器人有限公司(上海总部):

上海市闵行区七宝镇新龙路 500 弄虹桥万创中心 T1 栋 7 楼

捷勃特制造及技术服务中心:

上海市青浦区久业路 338 号 1 幢

捷勃特机器人东莞分公司:

广东省东莞市松山湖园区工业南路 14 号 9 栋 801 室

服务热线: 400-996-7588

网址: www.sh-agilebot.com